Seenaptec Access Platform

Administrator Guide

System / Platform: **Seenaptec SKD / Access Platform (SAP)**

Software Version: `v1.7.0`

Target Market / Language: International / English

Document Edition

The `INT` code denotes the international edition, and `EN` denotes the publication language (English). The timestamp indicates the UTC generation time of this PDF build.

`SAP-ADMIN-INT-EN-20261020T134507Z`

Seenaptec LLC

https://www.seenaptec.com

Table of Contents

Chapter 1. About This Document **4**
1.1. Purpose . . . 4
1.2. Structure . . . 4
1.3. Conventions . . . 4
1.4. Document Currency . . . 5

Chapter 2. System Purpose **5**
2.1. Object-Centric Access Control Model . . . 5
2.2. Problems Solved by the System . . . 5
2.3. Key Capabilities . . . 6

Chapter 3. Key Concepts and Glossary **6**
3.1. System Model . . . 6
3.2. Glossary of Terms and Abbreviations . . . 8

Chapter 4. How the System Works **11**
4.1. Server and Browser . . . 11
4.2. Overview of Service Interactions . . . 11
4.3. Access Request Processing . . . 13
4.4. Data Storage . . . 13
4.5. FAS Loop (Modbus TCP) . . . 13
4.6. Web Interface and API URLs . . . 14

Chapter 5. Technical Specifications **14**
5.1. Platform and Delivery Package . . . 14
5.2. Minimum System Requirements . . . 15
5.3. Operator Workstation . . . 15
5.4. Peripherals and Protocol . . . 15
5.5. Line Connection Types . . . 16
5.6. Limits and Redundancy . . . 16
5.7. Data Stores . . . 16
5.8. REST API and OpenAPI . . . 17
5.9. Implementation Technologies and External Dependencies . . . 17

Chapter 6. Installation **17**
6.1. Preparing for Installation . . . 17
6.2. Preparing the MQTT Broker . . . 18
6.3. Installing the Package . . . 18
6.4. Configuration . . . 18
6.5. Post-Installation Verification . . . 18
6.6. System Activation . . . 19
6.7. First Sign-In and Changing the Default Password . . . 19
6.8. Enabling HTTPS . . . 19
6.9. Upgrading the Package . . . 19

Chapter 7. Initial System Configuration **20**
7.1. Creating an OSDP Line . . . 20
7.2. Discovering and Registering Devices . . . 21
7.3. Creating an Access Object . . . 21
7.4. Creating a Person and Issuing a Pass . . . 22
7.5. Testing Access and Verifying Journal Events . . . 22
7.6. Further System Configuration . . . 22

Chapter 8. UI Basics **23**
8.1. Signing In . . . 23
8.2. Workspace Screen Layout . . . 23

8.3. Top Bar and Workspaces . . . 24
8.4. Left Area: Panels . . . 25
8.5. Right Area: Card . . . 26
8.6. Tool Panel . . . 27
8.7. “Events” Panel . . . 27
8.8. Verifying Command Execution — “Command Accepted Does Not Mean Completed” . . . 28
8.9. General UI Behavior Rules . . . 28
8.10. Practical Tips for Working with the Interface . . . 29

Chapter 9. System Overview **29**
9.1. How to Open the Page . . . 29
9.2. Status Cards . . . 30
9.3. Metrics and Additional Sections . . . 31
9.4. “Requires attention” List . . . 31
9.5. Automatic Refresh . . . 32
9.6. Tool Panel . . . 32

Chapter 10. Setup / Maintenance **33**
10.1. “Equipment” Panel . . . 33
10.2. Lines . . . 34
10.3. Discovering and Adding Devices . . . 35
10.4. Devices . . . 36
10.5. Bulk Operations . . . 38
10.6. Equipment Events and “Commissioning (LIVE)” Mode . . . 38
10.7. “Debug” Panel . . . 39
10.8. System Tools (Tool Panel) . . . 39
10.9. FAS Equipment (UniPOS Panels) . . . 42

Chapter 11. Access Objects **44**
11.1. Types of Access Objects . . . 44
11.2. “Access Objects” Panel . . . 45
11.3. Access Object Card . . . 46
11.4. Object Access Rules . . . 47
11.5. Device Bindings . . . 48
11.6. Rules Catalog: Conditions, APB Perimeters, and Quota Debit . . . 49
11.7. Production Calendar . . . 50
11.8. “Active Access” Panel . . . 50
11.9. “Events” Panel . . . 50
11.10. “Routes” Catalog . . . 51
11.11. “Presence tracking” Catalog . . . 51
11.12. FAS Objects (Fire Zones D, A, P) . . . 51

Chapter 12. Passes **53**
12.1. Core Principle: A Pass Is a Document . . . 53
12.2. Passes Panel . . . 54
12.3. Person (Pass Holder) . . . 56
12.4. Issuing a Pass and a Pass Request . . . 57
12.5. Access Templates and Restricted Issuance . . . 58
12.6. Identifiers Panel . . . 58
12.7. Events Panel . . . 62
12.8. Custom Personnel Fields . . . 62
12.9. Importing Personnel from a File . . . 62
12.10. Auto-Cleanup . . . 64

Chapter 13. Journals **64**
13.1. Journal Architecture . . . 64
13.2. Journals Panel: Searching History . . . 65

13.3. Event Card . . . 66
13.4. Alarms . . . 66
13.5. System Journal Integrity Verification . . . 67
13.6. Archiving . . . 68
13.7. Viewing Archives . . . 68
13.8. Reports . . . 68

Chapter 14. Security **69**
14.1. Permission Model (RBAC) . . . 69
14.2. Operators Panel . . . 70
14.3. Roles . . . 71
14.4. Sessions Panel . . . 72
14.5. Secrets (Keys) . . . 72
14.6. Security Settings . . . 72
14.7. Events Panel: Platform Audit . . . 73
14.8. Reader Workstations (Desktop Reader / Card Agent) . . . 73
14.9. Card Profiles . . . 75

Chapter 15. Common Scenarios **77**
15.1. Issue a Visitor Pass by Template . . . 77
15.2. Pass Request and Issuance . . . 77
15.3. Revoke a Pass or Block a Person . . . 77
15.4. Check Who Opened a Door . . . 78
15.5. Acknowledge an Alarm . . . 78
15.6. Connecting a UniPOS Fire Panel and Importing Configuration . . . 78
15.7. Reset Anti-Passback Presence or a Cabinet Session . . . 78
15.8. Replace a Faulty Device . . . 79
15.9. Update Device Firmware . . . 79
15.10. Create a Backup and Restore . . . 79
15.11. Filter Events for a Time Period . . . 80
15.12. Verify Journal Integrity and Inspect an Archive . . . 80

Chapter 16. Troubleshooting **80**
16.1. Web Interface Does Not Open . . . 80
16.2. Interface Opens, but Data Does Not Update . . . 81
16.3. Line in “Offline”, “Possible break”, or “Inactive” State . . . 81
16.4. Device Does Not Respond or Is Incomplete After Discovery . . . 81
16.5. Access Is Denied Even Though the Pass Is Active . . . 81
16.6. No Access Event Appears at All . . . 82
16.7. Redundancy Ring Is Not Normal . . . 82
16.8. “Journal integrity violation” Alarm . . . 83
16.9. Forgotten Administrator Password . . . 83
16.10. FAS Panel and Its Devices in “OFFLINE” State . . . 83

Chapter 17. License: Activation and Lifecycle **83**
17.1. How Licensing Works . . . 83
17.2. Activation Key and Two Identifiers . . . 84
17.3. Activating a Clean System . . . 84
17.4. Expired License and Renewal . . . 84
17.5. Software Update and the 14-Day Grace Period . . . 85
17.6. Migration, Restore, and Reactivation . . . 85
17.7. Support Code (Installation Fingerprint) . . . 85
17.8. Access Recovery Using a Vendor Code . . . 86
17.9. Common Situations . . . 87

Chapter 18. API Integration **88**
18.1. Specifications and Swagger UI . . . 88

18.2. External System Authentication . . . 88
18.3. Permission Model . . . 89
18.4. Permissions for Common Integration Tasks . . . 89
18.5. Implementation Notes . . . 89

Chapter 19. Company Details and Contacts **90**
Skolkovo Innovation Center . . . 90

Chapter 1. About This Document

1.1. Purpose

This guide is intended for installing, initially configuring, administering, and operating the **Seenaptec Access Platform (SAP)** access control system.

The document describes the system architecture and core concepts, working with the web interface, configuring equipment and access objects, managing passes, handling events and alarms, backing up and restoring the system, troubleshooting faults, and integrating via the REST API.

This guide does not replace:

- site design and working documentation; equipment mounting and electrical wiring manuals;
- documentation for RS-485 converters and peripheral devices; the REST API specification;
- information security regulations of the operating organization.

This guide is addressed to system administrators and operators. The approximate mapping between roles and chapters is shown below:

Reader	Key chapters
System Administrator	4–6, 9–10, 14, 17
Operations Engineer	4–5, 7, 9–10, 16
Access Administrator	3, 9, 11, 15
Pass Office Operator	3, 9, 12, 15
Security Operator	8–9, 13, 15
Integrator	3–4, 18

When getting acquainted with the system for the first time, read Part I in full: it establishes the terminology used throughout the remaining chapters.

1.2. Structure

- **Part I. Introduction to the System** (Chapters 1–5) — purpose, key concepts and glossary, how the system works, and technical specifications.
- **Part II. Installation and First Run** (Chapters 6–7) — server installation and a step-by-step initial configuration walk-through.
- **Part III. Interface and Daily Operations** (Chapters 8–14) — general interface principles, the “System Overview” start page, and descriptions of the workspaces: “Setup / Maintenance”, “Access Objects”, “Passes”, “Journals”, and “Security”.
- **Part IV. Scenarios and Reference** (Chapters 15–19) — step-by-step common scenarios, troubleshooting, license and its lifecycle, API integration, and company details and contacts.

1.3. Conventions

- The → symbol indicates sequential navigation across workspaces, panels, sections, or interface commands.
- Names of workspaces, panels, buttons, and other screen elements are shown in quotation marks and match the labels in the interface: the “Passes” workspace, the “Equipment” panel, the “Save” button.

- Commands, addresses, file names, and parameter values are set in a monospaced font: `sudo systemctl status srv_ctrl.service`, `http://localhost:8080/app/`, `seenaptec_acs.yaml`.
- Text in angle brackets indicates a placeholder that must be replaced with actual data. Do not type the angle brackets themselves. Example: `http://<server-address>:8080/app/`
- Step-by-step procedures are numbered; the expected outcome is stated after the steps.
- Callouts beginning with **"For users familiar with traditional ACS"** compare the system model with familiar concepts from conventional access control systems — access levels, cardholder databases, and access point groups.
- Screenshots were captured on a reference test bench; the server address and port shown in the images may differ from yours.

1.4. Document Currency

This guide corresponds to Seenaptec Access Platform (SAP) **version v1.7.0** (October 2026). The installed system version is displayed in the "Product" card on the "System Overview" start page (Section 9.3). All procedures, interface element names, and stated features have been verified for this release of the product. Everything described in the main chapters is implemented and available in the delivery of this version.

If you are using a different version, refer to the corresponding edition of this guide. Interface element names, available features, and procedure steps may differ.

Before performing any procedures, compare the installed system version with the version stated on the title page. For other versions, use the appropriate edition of the guide.

Chapter 2. System Purpose

2.1. Object-Centric Access Control Model

Seenaptec Access Platform (SAP) combines in a single web interface an Access Control System (**ACS**) and a Fire Alarm and Automation System (**FAS**, based on addressable UniPOS IFS7000 series panels over Modbus TCP). At the center of the access control model is the protected access object: a door, turnstile, server cabinet, or any other point where access must be restricted and monitored, while the fire safety domain is organized around logical FAS zones for detection (`D`), alarm/notification (`A`), and automation (`P`).

A physical protected asset is represented in the system by a logical entity — an **access object**. The object type defines its operating modes, available rules, and the set of equipment components with which it interacts. As a result, access permissions and behavior settings do not depend directly on a specific reader, relay, or hardware device instance.

Configuration is performed "from the object": first, the administrator defines what is being protected and how the access object should operate; next, the administrator assigns who may gain access, using which identifiers, when, and under what conditions.

Permissions are assigned both to individual access objects and to groups of objects combined by tags. A pass links a person, identifiers, validity dates, and permitted objects, while the object's own settings define its operating mode and additional restrictions. The final access decision is made based on the pass data, the state of the object, and the active rules.

For objects that support prolonged access, the system tracks access sessions. For example, an engineer's work at a server cabinet can be recorded as a session with a start event, periodic activity confirmations, and a completion event.

Access decisions, configuration changes, operator actions, and alarms are recorded in the event journal. Stored records cannot be modified through the web interface or REST API; the integrity of the event sequence can be verified using built-in system tools.

2.2. Problems Solved by the System

Seenaptec Access Platform (SAP) provides administrators and operators with the information and tools needed to manage access, monitor and control fire alarm and automation equipment, supervise hardware health, and investigate events in a single interface.

The system makes it possible to determine:

- which ACS lines, peripheral devices, and components, as well as FAS control panels, signaling loops, and addressable FAS devices, are registered, whether communication with them is established, and whether any technical faults are present;
- which access objects have been created, what types they belong to, which equipment they interact with, and what operating mode they are currently in, as well as the current state of fire FAS zones (`D`, `A`, `P`);
- which persons, identifiers, and passes are registered, which objects they are granted access to, and under what time intervals, schedules, and restrictions that access applies;
- what decision was made during an access attempt and which data or rules caused access to be granted or denied;
- which persons are currently recorded as present inside monitored perimeters and which access sessions are active;
- which access, audit, and technical health events have been recorded and which alarms require operator attention;
- who made changes to system configuration, access permissions, and system data, when those changes were made, and what exactly changed;
- which operator and administrator accounts exist, which roles and permissions are assigned to them, and which operator sessions are currently active.

A detailed description of the "System Overview" page and the corresponding workspaces is provided in Chapters 9–14.

2.3. Key Capabilities

The current version of Seenaptec Access Platform supports:

- **equipment** — OSDP connectivity via USB and IP gateways, automated device discovery supporting Seenaptec SecureBus (SSB), configuration of internal parameters for AGRG devices, health and functional testing, encrypted communication via OSDP Secure Channel, line redundancy, firmware updates for AGRG devices, and bulk operations;
- **access objects** — object types "Door (entry/button)", "Door (entry/exit)", "Server cabinet (1 door)", "Server cabinet (2 doors)", "Turnstile", and "Barrier", assignment of device components, identification modes (card, card + PIN, confirmation), access rules, perimeters with anti-passback (APB), blocking conditions, routes, presence zone marking, access time templates and production calendar, and the "Active Access" panel;
- **fire alarm and automation (FAS)** — connection of addressable UniPOS IFS7000 series fire panels over Modbus TCP, automated composition discovery of system blocks, loops, and addressable detectors/modules, import of custom labels and zone bindings from UniPOS XML configuration files, and real-time monitoring and operator control of detection (`D`), notification (`A`), and automation (`P`) zones;
- **pass office** — persons with custom fields and photos, identifiers, passes with objects and tags, routes, validity periods, access time modes, and passage limits, access templates and identifier templates, pass requests, and automated lifecycle management;
- **journals and alarms** — a unified event journal covering access, audit, and technical health events, an event card showing participants and "before/after" field diffs, alarms requiring a mandatory comment upon acknowledgment, reports, and archiving with integrity verification;
- **platform security** — operators, roles with domain- and level-based permission scoping, operator sessions, a secrets vault, and platform audit logging;
- **integration** — a comprehensive REST API covering all system domains, complete with OpenAPI specifications and an embedded Swagger UI.

Access to the system is provided via a web interface in a standard browser; no separate workstation software or desktop client installation is required. Server installation is described in Chapter 6, and system architecture is covered in Chapter 4.

Chapter 3. Key Concepts and Glossary

This chapter describes the system data model — how access objects, subjects, passes, and equipment are related — and provides a glossary of terms used throughout this guide and the web interface.

3.1. System Model

Seenaptec Access Platform (SAP) is built around two chains of entities that meet at the access decision point.

Physical layer (equipment level):

- **Controller (Server)** — the central component of the system; the computing node that runs the platform services and stores system data. The server communicates with devices over OSDP lines through local or network RS-485 converters.

- **Line (OSDP line)** — a communication channel configured in the system that uses the OSDP protocol between the server and addressable devices on an RS-485 bus. A line uses a USB or network converter and can include a primary and a backup communication loop.
- **Peripheral Device (PD)** — an addressable OSDP peripheral device registered on a line. A device provides functional components that can be used by access objects (doors, turnstiles, server cabinets, etc.).
- **Device component** — a functional unit of a device available in the system for binding to an access object: a reader, a contact (input), a relay (output), etc.
- **Fire Alarm Control Panel (FAS panel)** — an addressable UniPOS IFS7000 series fire panel connected to the server over Modbus TCP, hosting system blocks (`PSU`, `BIO`, relay/input blocks) and ring **signaling loops (Loops)** with **addressable FAS devices** (fire detectors, input/output modules, and automation control devices `FPE`).

Logical layer (access control and fire safety level):

- **Pass holder / Person** — a system entity representing an access subject, such as an employee or visitor. Identifiers and passes can be associated with a person.
- **Identifier** — a credential value or physical token presented during an access request and linked to a person, such as a card or PIN.
- **Pass** — an electronic document that defines a person's access permissions. A pass contains permitted access objects or tags, a validity period, an access time profile, and usage restrictions.
- **Access Object (AO)** — a logical representation of a protected point, such as a door, turnstile, or server cabinet. The object type defines its operating modes, rules, and functional slots.
- **Tag** — a named label assigned to an access object. When a tag is included in a pass, the permission extends to all access objects bearing that tag.
- **FAS Zone (`D`, `A`, `P`)** — a logical zone of a fire panel grouping detectors, sounders, and executive modules: detection zone (`D`, Detection), notification zone (`A`, Alarm/Notification), and automation zone (`P`, Automation/Protection).

Linking Equipment to Access Objects

Device components are bound to the functional slots of an access object. For example, a door's slots can be bound to a reader, a lock relay, an exit button, a door contact sensor, and so on.

An access object and the permissions granted to it do not depend on a specific hardware device instance. When a device is replaced, the access object, passes, and event history are preserved; only the component bindings need to be restored or confirmed.

An access decision is made at the moment an identifier is presented at a reader bound to an access object. The system locates the identifier's owner, checks their active passes, permissions for the object (directly or via a tag), and access time mode, as well as the object's own rules — conditions, anti-passback (APB), and confirmation requirements — and then grants or denies access. The outcome and its reason are recorded in the event journal.

Entity Relationship Diagram

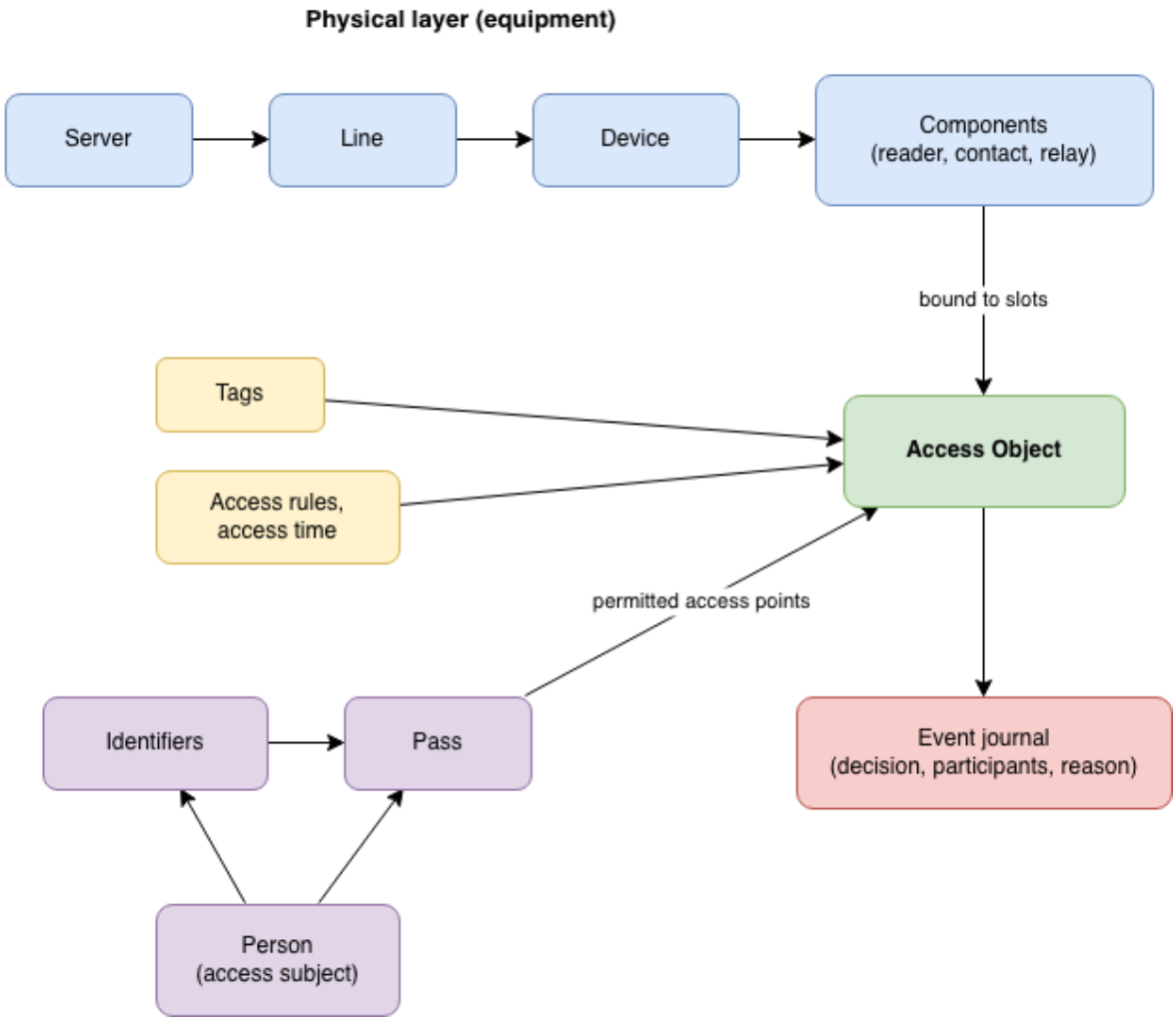


Figure 3.1 — General entity relationship diagram

Object-Side Rules and Restrictions

In addition to the permissions granted by a pass, the access decision is governed by the settings of the access object:

- **access time** — three duration profiles (normal, reduced, and extended access time) and the production calendar;
- **access rules** — conditions that a subject is required to have or not have (for example, a security clearance or safety briefing), and perimeters with anti-passback (APB);
- **identification mode** — card only, card + PIN, or card + confirmation (dual control / escort).

Operators and Persons

Users of the web interface are registered in the system as **operators**. Each operator is assigned a role and corresponding permissions.

An operator and a person are distinct entities. For example, an employee can both work in the web interface and use a pass to walk through a door, but their operator account and person record are stored and managed separately. Operator actions are recorded in the audit journal.

3.2. Glossary of Terms and Abbreviations

Term	Description
Emergency offline opening	Autonomous unlocking of an access object by an AGRG device when communication with the server is lost: triggered by an emergency release button or emergency cards. Configured in the access object card (Section 11.3).
Addressable FAS Device	A fire detector (smoke, thermal, combined, gas, manual call point), input/output module, or automation control device (`FPE`) on a signaling loop or built-in block (`BIO`/relay block).
APB (anti-passback)	A perimeter rule: a subject who has entered a perimeter cannot enter again without first exiting (and vice versa). The mode can be hard (denies access) or soft (logs a journal event only).
Desktop reader workstation	A workstation equipped with a desktop card reader: an agent application running on the operator's computer and connected to the server. Used for reading, issuing, and initializing cards (Section 14.8).
Audit	A journal category: data and configuration change events showing who changed what and when, with a list of "before/after" field values.
Access time	The time interval during which access is permitted at an access object. Each object has normal, reduced, and extended access time; the applied profile is determined by the pass and the object's access time template.
Event journal	A time-ordered stream of access, audit, and technical health records. Stored records cannot be edited and are protected by cryptographic hashing; the only mutable layer is operator marks.
Pass request	A draft pass: reserves a pass number and holder data but does not grant access until the pass is issued. Supports separation of duties: one group of operators prepares requests, while the pass office approves the pass and assigns access objects.
FAS Zone (`D`, `A`, `P`)	A logical zone of a FAS panel: detection zone (`D`, Detection), notification zone (`A`, Alarm/Notification), and automation zone (`P`, Automation/Protection).
Identifier	A card, PIN code, or other credential presented to identify an access subject. Tracked in the "Identifiers" panel regardless of whether a pass has been issued.
Card (detail pane)	The right-hand pane of a workspace displaying details of the selected entity: controller/server, line, peripheral device, access object, pass, person, or journal record.
Device component	A functional unit of a peripheral device — reader, contact (input), or relay (output) — that can be bound to a slot of an access object.
Line (OSDP line)	An OSDP communication channel between the server and a group of peripheral devices sharing a physical RS-485 bus; connected via a USB or network converter.
Route	A named set of transit access objects. When assigned to a pass, a route grants access to all of its points without distinguishing between entry and exit. The route catalog is described in Section 11.10.
Mark	An operator annotation placed on a journal event for investigation purposes. Adding and removing a mark is itself recorded in the journal.
Access Object (AO)	A point where access is controlled — a door, turnstile, cabinet, or storage area. In the system, it is a logical entity composed of functional slots whose behavior is defined by the access object type; permissions and rules are assigned to the object.
Operator	A user account for working in the system web interface. An operator's permissions are determined by their assigned role.
Panel	A pane on the left side of a workspace containing a list, tree, or feed: for example, the "Equipment" panel or the "Events" panel.
Tools drawer (Toolbar)	A slide-out drawer along the right edge of the screen containing directories and settings for the current workspace: templates, rules, reports, and system operations. Opened via the gear button in the bottom-right corner.
Perimeter	A named zone used for APB rules and presence tracking: object readers define the perimeter's entries and exits, and the system tracks which subjects are currently inside.
Pass holder / Person	A record representing an access subject: an employee, visitor, or bearer pass holder. A person can have identifiers, passes, a photo, and custom fields.
Access rule	A condition or restriction assigned to an access object: requiring the presence or absence of a condition on the subject, or including the object in an APB perimeter.
Fire Alarm Control Panel (FAS panel)	An addressable UniPOS IFS7000 series fire panel connected over Modbus TCP. Hosts system blocks (`PSU`, `BIO`, relay/input blocks), ring signaling loops, and logical fire zones.
Production calendar	A calendar of working days, weekends, and public holidays taken into account by access time templates.

Term	Description
Card profile	A named definition of a card's secure memory area: technology (MIFARE Classic/Plus, DESFire), read and write keys from the secrets vault, and code location. The card profile library is in the "Security" workspace (Section 14.9).
Pass	A set of access terms assigned to a person: access objects, tags, or routes, validity period, access time mode, permitted identifiers, and restrictions such as a passage quota.
Workspace	A top-level section of the web interface. There are five workspaces: "Passes", "Access Objects", "Journals", "Setup / Maintenance", and "Security". In the current UI version, workspaces are also referred to as "tools".
Role	A set of operator permissions defining access levels across system domains (equipment, access objects, persons and passes, security, journal).
Secret	A named protected value stored in the "Security" workspace vault: a backup password, OSDP key, card key, or HTTPS certificate. A secret's value cannot be read back from the interface — it can only be referenced and used.
Controller (Server)	The central node of the system; in the equipment tree, it is the root entry under which lines are organized.
Access session	A prolonged access episode with a start, periodic confirmations, and completion — primarily for server cabinets: who opened it, with which pass, and when the last confirmation occurred. Active sessions are shown in the "Active Access" panel.
Operator session	An operator's working session in the web interface from sign-in to sign-out; automatically terminated after prolonged inactivity. Active sessions are visible in the "Security" workspace.
ACS	Access Control System.
Slot (functional slot)	A functional element of an access object — reader, lock, sensor, or button. The set of slots is defined by the access object type; a device component is bound to each slot. Slots can be mandatory or optional.
Event	A single journal record: an access attempt, configuration change, equipment state transition, or service/operator action.
Access subject	An entity — a person, vehicle, or asset — subject to access control; represented in the system as a person.
Tag	A label grouping multiple access objects for permission assignment: a pass can grant access by tag instead of listing each object individually.
Active Access	A panel in the "Access Objects" workspace showing who is currently inside perimeters and which access sessions are open.
Access object type (Access object template)	A predefined set of slots, interaction rules, and operating modes that determines the behavior of an access object. Selected when creating an object; the list of types is given in Section 11.1.
Alarm	A journal event requiring operator acknowledgment. Active alarms are indicated by the bell icon in the top bar of the interface and must be acknowledged with a mandatory comment.
Condition	A named attribute of a subject used by access rules: a condition can be set upon passage, or its presence or absence can be required for access.
Peripheral Device (PD)	An addressable peripheral device on an OSDP line, consisting of components — readers, contacts, and relays. The device does not make normal access decisions; decisions are made by the server.
Access template	A pass preset defining validity period rules, objects, tags, access time, and passage quota for rapid issuance of standardized passes.
Identifier template	A preset for issuing cards: defines the card kind and validity period rule.
Signaling Loop (Loop)	A ring addressable line of a FAS panel connecting fire detectors and input/output modules (up to 210 addressable devices per loop).
Identifier issuance (encoding)	Writing a credential code into a card's secure memory area on a desktop reader while simultaneously creating the identifier record in the system (Section 12.6).
MQTT	The messaging protocol used between platform services. Requires a separate MQTT broker in the deployment environment.
OpenAPI / Swagger UI	A machine-readable description of the REST API and an interactive web page for browsing and invoking API methods; available at `http://<host>:8080/api-docs/`.
OSDP	Open Supervised Device Protocol — an open communication protocol for access control peripheral devices developed by SIA, operating over RS-485.

Term	Description
OSDP Secure Channel	An encrypted communication channel with an OSDP peripheral device; supports key installation and rotation.
REST API	An HTTP-based programmatic interface for managing configuration, access data, and journals; see Chapter 18.
Seenaptec Access Platform	The access control platform described in this guide.
SQLite	An embedded relational database engine storing data in `*.sqlite` files; used as the built-in data store so no external database server is required.
SSB (Seenaptec SecureBus)	A proprietary operating mode for supported OSDP devices providing automated discovery, address assignment, and line baud rate negotiation.
systemd	The Linux service management subsystem; the platform runs under the `srv_ctrl.service` unit.

Chapter 4. How the System Works

4.1. Server and Browser

The server components of Seenaptec Access Platform (SAP) are installed on a host running a recommended Linux distribution. Host hardware and software requirements are listed in Section 5.2.

Operators interact with the system through a supported web browser. No separate client software installation is required: the web application is loaded from the server and runs directly in the operator's browser. Workstation requirements are listed in Section 5.3.

All operators use a single web interface URL. The available workspaces and operations are determined by each operator's assigned roles and permissions. Default URLs and ports are listed in Section 4.6.

When operating the system from remote workstations, enabling HTTPS is recommended. The procedure for enabling HTTPS is described in Section 6.8.

The server components (services) communicate with ACS peripheral equipment and FAS panels, implement core platform logic, and store configuration, access data, fire alarm data, and the event journal. The service architecture is described in Section 4.2, the access decision flow is explained in Section 4.3, and the FAS operating loop is covered in Section 4.5.

4.2. Overview of Service Interactions

The platform consists of several services, each responsible for a dedicated domain. All services are installed from a single package and started automatically.

Service	Purpose
`srv_ctrl`	Process supervisor: reads the configuration, starts platform services, monitors their health, and restarts them on failure. Managed by systemd as `srv_ctrl.service`.
`scud-server`	**Application server**: serves the web interface, REST API, event journal, equipment, access, and FAS configuration, and OpenAPI documentation.
`srv_osdp`	**OSDP line N server**: handles device polling, discovery, commands, and events. One instance runs per line; when line redundancy is enabled, a single instance manages both ports of the ring (primary and backup).

Service	Purpose
`srv_ops`	**FAS panel N server**: polls an addressable UniPOS IFS7000 series fire panel over Modbus TCP, performs automated composition discovery, executes zone and device commands, and publishes states and events. One instance runs per enabled FAS panel.
`srv_behavior`	**Behavior server**: executes access object logic, access decisions, sessions, perimeters, and object states.
`srv_discover`	**Device discovery server**: discovers network RS-485 converters during line configuration.

Services exchange operational messages — events, commands, and states — through an **MQTT broker**. The broker is an external dependency: it is not bundled inside the platform package and must be deployed separately in the host environment (typically Eclipse Mosquitto). The web browser does not communicate via MQTT; the web interface exchanges data exclusively with `scud-server` over HTTP/HTTPS.

Architecture Overview Diagram

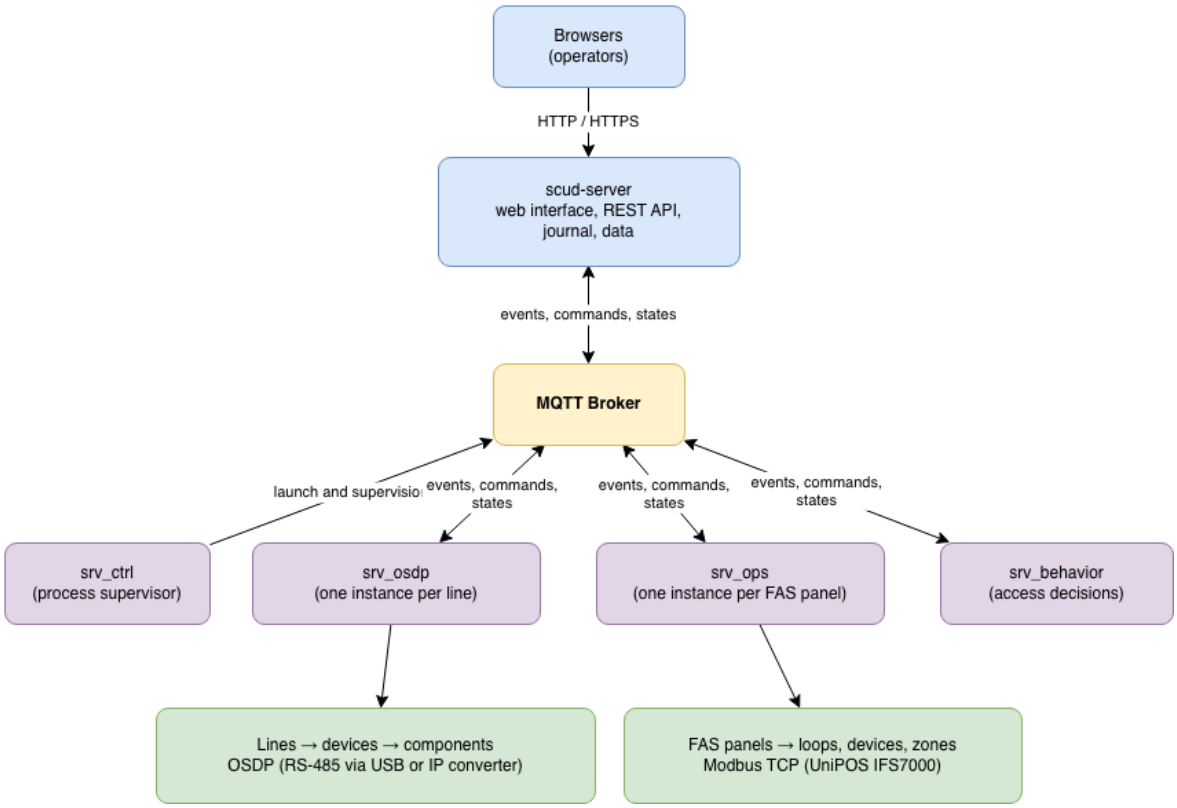


Figure 4.2 — Overview diagram of service interactions

The diagram is intentionally high-level: for day-to-day operations, it is sufficient to understand that the browser communicates with the central application server, which in turn controls OSDP lines and receives events via internal messaging. Service status is visible in the web interface under the “Service Management” section of the “Setup / Maintenance” workspace.

4.3. Access Request Processing

Below is the standard access request flow when a card is presented.

A reader reads the card code, and the peripheral device transmits the event over the OSDP line to the `srv_osdp` instance servicing that line. Based on the reader's binding configuration, the system identifies the associated access object and passage direction.

`srv_osdp` publishes the access request message via the MQTT broker to `srv_behavior`. The behavior server sequentially:

1. identifies the person linked to the identifier;
2. verifies the status of the person and the identifier;
3. locates matching active passes;
4. verifies permission for the access object directly or via a tag;
5. checks the pass validity period, access time schedule, and usage limits;
6. evaluates the access object's operating mode and rules, including access conditions, APB, and confirmation requirements.

Based on these checks, `srv_behavior` reaches a decision: grant or deny access. If an additional factor is required (such as a PIN or escort confirmation), the system waits for it to be presented and forms the final decision once all mandatory checks are complete.

The decision is sent via the MQTT broker back to the `srv_osdp` instance. When access is granted, `srv_osdp` sends the appropriate actuator/indication command for the object's type and mode to the peripheral device.

Door session and group passage. On a door equipped with a door position sensor, opening the door leaf starts a door session: while the leaf remains open, card presentations by other persons are registered as their own passages — with full permission verification, APB tracking, and pass quota deduction, but without re-energizing the lock relay; a PIN is not requested during such registration because the door is already open. Presenting a card without permission for the object during an open door session records a critical "Access denied" event and raises an "Unauthorized passage" alarm. If the door was opened with confirmation (dual control or guest escort), the session remembers the confirming person: subsequent persons marked "Requires escort confirmation" and guests of the same escort are registered without requiring a new confirmation, whereas presenting a card for a person not covered by the session's confirming person is rejected with the same alarm. Presenting the confirming person's own card again registers their own passage.

Operator commands issued from the web interface are executed **asynchronously**. An accepted command means the request has been queued, not yet completed on hardware; the actual outcome is confirmed by a state change on the peripheral device or a corresponding event in the journal.

The access decision, its reason, and associated participants are recorded in the event journal. Command execution confirmations and object state transitions are logged as separate events as telemetry arrives from the equipment.

System behavior when encountering an unknown identifier, a missing active pass, unavailable server services, or lost communication with a device is detailed in Chapter 16.

4.4. Data Storage

System data is stored on the server in embedded **SQLite** database files — no external database server is required. The files are partitioned by functional domain: equipment and access object configuration, the FAS subsystem (panels, loops, devices, and zones), persons and passes, the event journal, operators and security settings, and runtime access state. This separation simplifies backup and migration: the built-in backup utility (Chapter 10) saves and restores these files together. The event journal follows an append-only design: stored records cannot be modified, making it suitable for auditing and integrity verification (Chapter 13).

4.5. FAS Loop (Modbus TCP)

The Fire Alarm and Automation subsystem (**FAS**) operates alongside the ACS loop and is served by dedicated instances of **`srv_ops`** (displayed in the UI as **"FAS panel N server"**). The central application server launches one isolated `srv_ops` process for each enabled UniPOS IFS7000 series FAS panel.

The FAS operating loop works in four stages:

1. **Connection and automated composition discovery.** The `srv_ops` instance establishes a Modbus TCP connection to the UniPOS panel (port `502` and `Unit ID = 255` by default), reads the device identification passport using function `0x2B / 0x0E` (model `7000M`, `7000-1MC`, or `7000M-1L`, serial number, firmware version), and upon a composition scan command automatically discovers installed system blocks (`PSU`, `BIO`, relay/input blocks), ring signaling loops, addressable FAS devices, and configured zones (`D`, `A`, `P`).
2. **Continuous state polling.** At the configured polling interval (1000 ms by default), `srv_ops` polls the status registers of system blocks, loops, addressable devices, and zones using functions `0x03` / `0x04`, and reads new text archive records from the panel's internal log using function `0x14`.
3. **Publishing states and events over MQTT.** Whenever status bit flags change (fire alarm, pre-alarm, warning or activation of notification and automation zones, fault, bypass/disable, test mode, or communication loss), `srv_ops` publishes the updated state and a state-delta event through the MQTT broker to `scud-server`. When TCP communication is lost, the panel and all its child elements transition to an explicit `OFFLINE` state in the UI.
4. **Unified journal and permission enforcement.** The central `scud-server` service ingests states and events from `srv_ops`, updates cards in "Setup / Maintenance" (Section 10.9), "Access Objects" (Section 11.12), and "System Overview" (Chapter 9), records events in the unified journal under the "FAS" category (Chapter 13), and enforces operator permissions subject to the FAS subsystem ceiling (Chapter 14).

4.6. Web Interface and API URLs

The web interface and service endpoints of Seenaptec Access Platform (SAP) are served by `scud-server`.

In the addresses below, replace `<server-address>` with the IP address or hostname of the server. Do not type the angle brackets. When accessing the system directly from the server host, you can use `localhost`.

By default, HTTP uses port `8080`.

Resource	Purpose	Default URL
Web interface	Operator and administrator workplace	`http://<server-address>:8080/app/`
Health check	Application server readiness probe for monitoring or diagnostics	`http://<server-address>:8080/health`
Swagger UI	Interactive browsing and invocation of REST API methods	`http://<server-address>:8080/api-docs/`
API documentation	OpenAPI specifications and accompanying technical documentation	`http://<server-address>:8080/docs/`

HTTP may be used for initial setup within a trusted network. For production operation and connections from remote workstations, HTTPS should be configured.

Once HTTPS is enabled, the web interface is available by default at:

`https://<server-address>:8443/app/`

Available resources, ports, and authentication rules depend on the server configuration. Do not expose the `/health`, `/api-docs/`, and `/docs/` service endpoints to untrusted networks without appropriate security controls and network restrictions.

Instructions for configuring HTTPS are provided in Sections 6.8 and 10.8.

Chapter 5. Technical Specifications

A reference summary for engineers and technical specialists. Step-by-step installation procedures are provided in Chapter 6, and commissioning procedures are in Chapters 7 and 10.

5.1. Platform and Delivery Package

Seenaptec Access Platform (SAP) is installed on a host running a supported Linux distribution. The software components are delivered as a `.deb` package for `amd64` and `arm64` architectures.

Parameter	Value
Operating system	Debian-family Linux distributions using `systemd` and the APT package manager (Debian, Ubuntu, Astra Linux, Linux Mint)
Installation package format	`.deb`
Supported architectures	`amd64`, `arm64`
Service manager	`systemd`
Primary service unit	`srv_ctrl.service`
Application binaries	`/opt/skd`
Configuration file	`/etc/skd/seenaptec_acs.yaml`
Service credentials	`/etc/skd/credentials/`
System data directory	`/var/lib/skd`
Default HTTP port	`8080`
Default HTTPS port	`8443` (when HTTPS is enabled)
External dependency	MQTT broker (installed separately)

The `/opt/skd` directory contains the immutable application files from the package. Configuration files, service credentials, and system databases are stored separately from the application binaries.

During a standard package upgrade, the following are preserved:

- the configuration file `/etc/skd/seenaptec_acs.yaml`;
- service credentials in `/etc/skd/credentials/`;
- system data in `/var/lib/skd`.

Before upgrading, create a backup of your system data and configuration as described in the backup and restore section.

5.2. Minimum System Requirements

Minimum server configuration for deploying the platform:

Component	Minimum requirement
Processor (ARM, tested)	4 cores Cortex-A76 class, 2.4 GHz, 64-bit ARMv8 — e.g., Broadcom BCM2712 (Raspberry Pi 5)
Processor (x86)	Modern-generation Intel Core i3, 4 cores, or comparable AMD processor (e.g., Ryzen 3)
RAM	2 GB
Disk storage	8 GB (journal storage requirements depend on the archiving policy)
Network	10/100 Mbps Ethernet. Network connectivity is mandatory: the system is client-server — the web UI and MQTT messaging run over the network

5.3. Operator Workstation

Parameter	Value
Workstation client	Web browser; no client software installation required
Supported browsers	Google Chrome, Firefox, Yandex Browser, Microsoft Edge, Safari (current releases)
Protocol	HTTP; HTTPS with certificate upload via the web interface

5.4. Peripherals and Protocol

Element	Description
Device protocol	OSDP; certain extended features require the Seenaptec SecureBus (SSB) extension

Element	Description
Communication security	OSDP Secure Channel: initial pairing using the default key (SCBK-D), installation of an operational key, and automatic key rotation mode (hardware-dependent)
Seenaptec SecureBus (SSB)	Automated discovery of compatible AGRG OSDP devices on a line: serial-number discovery, address assignment, baud rate negotiation (hardware-dependent), and internal device settings
Firmware update	Over-the-bus OSDP device firmware update from the web interface (OSDP File Transfer), including checksum and version verification after flashing
FAS protocol and equipment	**Modbus TCP** (port `502` by default); addressable **UniPOS IFS7000 series** fire alarm control panels (`7000M`, `7000-1MC`, `7000M-1L`)
FAS zones and configuration	Logical detection (`D`), notification (`A`), and automation (`P`) zones; automated composition discovery over Modbus (`0x2B/0x0E`, `0x03`, `0x04`, `0x14`) and `.xml` configuration import from UniPOS software

5.5. Line Connection Types

The following adapter types can be used to connect OSDP lines:

Interface type	Connection description
`USB->485`	Local USB-to-RS-485 converter (serial port on the server)
`TCP->485 (Other)`	TCP connection to a generic Ethernet-to-RS-485 converter operating in TCP server mode
`TCP->485 (DR134)`	TCP connection to a USR-DR134 converter with model-specific handling

To discover network converters on the local network, the platform provides the “Network adapter configuration” tool (Chapter 10).

5.6. Limits and Redundancy

Metric	Value
Maximum devices per line	64 (OSDP addresses 0–63)
Service per line	One `srv_osdp` instance; when redundancy is enabled, the same instance manages both ring ports (primary and backup)
Line redundancy	Primary and backup port (communication loop) for a single RS-485 bus segment; automatic failover upon bus break or converter/channel loss
FAS panel cluster and loops	Up to 8 UniPOS IFS7000 panels in a cluster (`Cluster panel number` 1–8); up to 2/4 signaling loops per panel (up to 10 on modular `7000M` units), up to 210 addressable devices per loop
Service per FAS panel	One isolated `srv_ops` instance (“FAS panel N server”) per enabled panel

Line redundancy is implemented by connecting two adapters of the same type to opposite ends of the physical RS-485 bus. Redundancy is activated automatically when a backup port is specified in the line settings. Both ports are managed by a single `srv_osdp` instance: under normal conditions, polling runs through the primary port; if the bus is broken, devices on each segment are automatically routed through whichever port remains reachable. A detailed description of redundancy states is given in Chapter 10.

5.7. Data Stores

Data is stored in SQLite database files inside the server data directory, separated by domain:

Domain	Contents
Configuration	Controller/server, lines, peripheral devices, access objects, bindings, templates
FAS subsystem	UniPOS fire panels, system blocks, signaling loops, addressable devices, zones (`D`, `A`, `P`), custom names imported from `.xml`
Access data	Persons, identifiers, passes, tags, access templates
Journal	Access, audit, and technical health events (append-only)
Operators and security	Operator accounts, roles, sessions, secrets
Runtime state	Perimeter presence, active access sessions, counters

Backup and restore operations are performed using the built-in mechanism in the web interface (Chapter 10); a backup archive includes both the database files and the server configuration.

5.8. REST API and OpenAPI

Resource	Default URL (browser on server)
Health check	`http://localhost:8080/health`
Web interface	`http://localhost:8080/app/`
OpenAPI, interactive (Swagger UI)	`http://localhost:8080/api-docs/`
API specifications and documentation	`http://localhost:8080/docs/`

REST APIs are available for equipment, access data, journals, and security; access for external systems is governed by the same role-based permission model as operator access (Chapter 18). When connecting from another computer, replace `localhost` with the IP address or DNS name of the server; the port number may differ if customized in the configuration file.

5.9. Implementation Technologies and External Dependencies

The server components of Seenaptec Access Platform (SAP) are written in Go, while the low-level OSDP protocol stack is implemented in C.

Platform services are started and supervised by `systemd`.

To exchange commands, events, and states between services, an external MQTT broker is required. The broker is not bundled in the Seenaptec Access Platform installation package and must be installed separately. It can run either on the platform server itself or on another reachable network host.

MQTT broker requirements and setup instructions are provided in Section 6.2.

Chapter 6. Installation

This chapter describes how to install the server components of Seenaptec Access Platform (SAP) on Linux, verify system readiness, activate the system, and sign in to the web interface for the first time. Technical specifications of the delivery package are provided in Chapter 5; equipment commissioning is covered in Chapters 7 and 10; the license lifecycle is described in Chapter 17.

6.1. Preparing for Installation

Before installing the system, prepare the following:

- a server or workstation of `amd64` or `arm64` hardware architecture that meets the requirements in Section 5.2 and runs a supported Linux distribution using `systemd` and supporting `.deb` package installation;
- a user account with `sudo` privileges;
- the Seenaptec Access Platform (SAP) installation package matching the server architecture;
- an activation key obtained from the system vendor (without an activation key, the server components can be installed and started, but signing in to the web interface and configuring the system will be unavailable);
- a running MQTT broker reachable on the organization’s network;

- an operator or administrator workstation with a supported web browser for initial system configuration.

The MQTT broker is an external dependency and is not included in the platform installation package. It can run either on the Seenaptec Access Platform (SAP) server itself or on a separate network node. Broker requirements and setup instructions are provided in Section 6.2.

Connecting peripheral equipment requires USB-to-RS-485 or Ethernet-to-RS-485 converters. They can be connected after the platform packages are installed, during the commissioning stage.

6.2. Preparing the MQTT Broker

Platform services exchange operational messages through an MQTT broker (Chapter 4). Deploy a broker according to your organization's standards or using your Linux distribution's package manager; the standard choice is Eclipse Mosquitto:

```
sudo apt install mosquitto
```

The broker can run either on the same host (default address `tcp://localhost:1883`) or on a dedicated server. Make sure the broker is running and accepts connections from the host where the platform is being installed.

6.3. Installing the Package

Install the package matching your server architecture:

```
sudo apt install ./seenaptec-acs_<version>_amd64.deb
```

The package places files and registers the service as follows:

Item	Location
Application binaries (immutable files)	`/opt/skd`
Configuration file	`/etc/skd/seenaptec_acs.yaml`
Service credentials	`/etc/skd/credentials/`
Data directory	`/var/lib/skd`
systemd service unit	`srv_ctrl.service` (enabled and started automatically)

The services run under a dedicated system user `skd`, which is created during installation. The user is automatically granted permissions to access serial ports (the `dialout` group), so no manual permission setup is needed for USB converters. The device discovery server is granted the `CAP_NET_RAW` network capability.

6.4. Configuration

Platform settings are stored in `/etc/skd/seenaptec_acs.yaml`. For initial startup, it is typically sufficient to verify two parameters:

- `MQTT.location` — the MQTT broker address, for example `tcp://localhost:1883`;
- `SRV_BACK.combined.addr` — the HTTP service bind address and port, by default `:8080`.

Lines and peripheral equipment do not need to be declared in the configuration file — they are configured from the web interface (Chapter 10). After modifying the configuration file, restart the service:

```
sudo systemctl restart srv_ctrl.service
```

6.5. Post-Installation Verification

1. Check the status of the supervisor service:

   ```
   sudo systemctl status srv_ctrl.service
   ```

   The expected status is `active (running)`.

2. Check the readiness of the HTTP service:

   ```
   curl -i http://localhost:8080/health
   ```

An HTTP 2xx response is expected.

3. Open the web interface in a browser: `http://<server-address>:8080/app/`. On an unactivated system, the "System activation" screen should appear (see below).

If any step fails, refer to Chapter 16 ("the web interface does not open"): common causes include a stopped service, an unreachable MQTT broker, or a port that is already in use or blocked by a firewall.

6.6. System Activation

When accessing the web interface for the first time, instead of the sign-in screen, the system displays the **"System activation"** screen with the prompt "Enter the activation key received from the vendor. An admin operator with the default password will be created." and a single "Activation key" field.

The activation key is issued by the system vendor (how to obtain it and all license details are covered in Chapter 17). The key consists of six groups of four hexadecimal characters (digits and letters A–F), for example `0107-0000-002A-9F3C-1B77-D041`. You can enter or paste it with or without hyphens, in lowercase or uppercase — the form automatically normalizes the input. The "Activate" button becomes enabled once all 24 characters have been entered.

After successful activation:

- the system is bound to the license encoded in the key;
- the `admin` operator account is automatically created with the default password `123456` and full permissions across all system domains (**Super Administrator**);
- the standard "Sign in to SKD" screen is displayed with the notice: "System activated. Default login: admin / 123456 — change the insecure password after signing in."

If the key is rejected, the form explains the reason: "Invalid activation key. Check your input." (a typo or a key issued for another installation) or "Too many activation attempts. Retry in N s." (brute-force rate limiting).

6.7. First Sign-In and Changing the Default Password

Sign in with `admin` / `123456`. Because the default password is the same across all installations, the system treats it as insecure: **every sign-in with the password `123456` is recorded in the journal as an alarm event** — an alarm will be raised on every sign-in until the password is changed (Chapter 14). Change the password immediately after your first sign-in.

How the password change proceeds depends on the "Require changing administrator-assigned password" setting ("Security" workspace → "Security settings", Chapter 14):

- **setting enabled** — immediately after signing in, the "Change Password" screen is displayed: current password, new password twice, and the "Change password" button. You can proceed to work in the system only after changing the password; the new password cannot match the default password (`123456`) and must satisfy the password security policy;
- **setting disabled** (by default) — access to the system is granted immediately, but alarm events on sign-in will continue until the password is changed. You can change the password in the `admin` operator card: "Security" workspace → "Operators" panel → `admin` → password change field.

Additional operator accounts, roles, and password policy parameters are configured in the "Security" workspace (Chapter 14). On subsequent visits to the system, the standard "Sign in to SKD" screen with the "Login" and "Password" fields is shown.

6.8. Enabling HTTPS

By default, the web interface is served over HTTP. HTTPS can be enabled directly from the web interface without an external reverse proxy: in the "Setup / Maintenance" workspace, the tools drawer contains the "HTTPS access" section for uploading a TLS certificate and private key (the step-by-step procedure is provided in Chapter 10). Once enabled, the HTTPS interface becomes available at `https://<server-address>:8443/app/` alongside the HTTP interface.

6.9. Upgrading the Package

To upgrade the platform, install the new package over the existing installation using the same command as during initial installation. An upgrade preserves:

- the configuration file `/etc/skd/seenaptec_acs.yaml`;

- service credentials in `/etc/skd/credentials/`;
- system data in `/var/lib/skd`.

The service restarts automatically. Before upgrading, create a backup using the built-in backup tool (Chapter 10).

Certain major upgrades increment the activation key generation — in that case, a 14-day grace period begins after the upgrade, during which you must obtain and enter a new activation key from the vendor (Section 17.5). Routine updates are not affected: if the key generation has not changed, no license actions are required.

Chapter 7. Initial System Configuration

This chapter walks through a complete end-to-end example of initial configuration in Seenaptec Access Platform (SAP) — from creating an OSDP line to recording the first access event.

The walkthrough covers the following steps:

- creating an OSDP line;
- discovering and registering a peripheral device;
- creating an access object;
- binding device components to the access object;
- creating a person and issuing a pass;
- testing access and verifying the event in the journal.

Before you begin, make sure that:

- the system has been installed in accordance with Chapter 6;
- the system is activated;
- you are signed in under an operator account with the required permissions;
- the default password for the `admin` operator has been changed;
- the RS-485 converter is connected to the server and recognized by the operating system;
- peripheral devices are wired to the line and powered on;
- you know the baud rate configured on the devices on the line.

This example uses an RS-485 line connected via a USB-to-RS-485 converter and an AGRG peripheral device that provides components for controlling a door: a card reader, a lock relay output, and an exit button input.

Detailed descriptions of equipment, access objects, passes, and the event journal are provided in Chapters 10–13.

7.1. Creating an OSDP Line

Before starting, verify that the USB-to-RS-485 converter is connected to the server and recognized by the operating system.

1. Open the “Setup / Maintenance” workspace and go to the “Equipment” panel.
2. Click “Add” → “Line”.
3. In the “Add line” dialog, specify:

- **Adapter type** — `USB->485`;
- **Primary port** — the serial port of the converter, for example `/dev/ttyUSB0`;
- **Line number** — a unique line number in the system;
- **Baud rate** — the communication speed configured on the connected devices.

4. Click “Create”.

Result: the line is created and added to the equipment list.

Verification: the communication state of the line should change to “Active” and subsequently to “Inactive”. The “Inactive” state appears after a short period without traffic on the bus, which is normal for an empty line with no registered devices.

If the communication state shows “Offline”, check:

- the selected serial port;
- the configured baud rate;
- the connection and power supply of the converter;

- the wiring and power supply of the devices;
- the RS-485 bus termination and wiring parameters.

Troubleshooting recommendations are provided in Chapter 16.

7.2. Discovering and Registering Devices

Before starting discovery, make sure the line is in the “Active” or “Inactive” state, and that the connected devices are powered on and operating at the baud rate configured for the line.

1. In the “Setup / Maintenance” workspace, select the newly created line in the “Equipment” panel.
2. Start device discovery:

- to discover AGRG devices supporting SSB, click **“SSB scan on current line”**;
- to discover other OSDP devices, open the line card and select **“Address discovery”**.

3. In the dialog that opens, click **“Search”**.
4. Wait for discovery to complete. Discovered devices will appear in the table along with their available details, including serial number and OSDP address.
5. Review the list of discovered devices. If some devices were not found, check their power supply, wiring, addresses, and baud rate, then repeat the scan. During an SSB scan on a line with a large number of devices, some units may occasionally be missed on the first pass; if so, repeat the scan until all devices are discovered.
6. Select the checkboxes for the devices you want to add to the configuration and click **“Apply selection”**.
7. If a device is saved with the status “New”, open its card and click **“Acknowledge”** (Accept).

Result: the selected peripheral devices are added to the equipment tree.

Verification: registered devices display the configuration state “Configured” and the communication state “Active”.

If a device is not discovered or communication with it cannot be established, refer to the troubleshooting guidance in Chapter 16.

7.3. Creating an Access Object

Before creating an access object, make sure that the peripheral device is registered in the system, is online (“Active”), and its components are properly displayed.

1. Open the “Access Objects” workspace.
2. Click **“Add access object”**.
3. In the dialog that opens, specify:

- **Name** — for example, `Entrance A`;
- **Unique code** — optional project design code from your installation blueprints. Can be used for automated binding of access objects to equipment according to the project specification;
- **Type** — for example, `Door (entry/button)`.

4. Click **“Add”**.
5. Select the newly created access object and open the “Device bindings” section in its card.
6. Assign the corresponding peripheral device components to the object’s slots:

- **Reader** — the card reader;
- **Lock** — the relay output controlling the door lock;
- **Exit button** — the digital contact input to which the exit button is wired.

7. Click **“Accept changes”**.

Result: the access object is created and bound to the peripheral device components.

Verification: the access object displays the state “Configured”, the communication state “Active”, and the operating mode “Standby”. The object is ready to process access requests.

7.4. Creating a Person and Issuing a Pass

Before creating a person, prepare a physical credential (card, keyfob, or token) to issue to the person and determine the desired pass validity period.

1. Open the "Passes" workspace.
2. Click **"New person"**.
3. Enter the person's last name, first name, and middle name (if applicable); optionally upload a photo, select a custom fields template, and fill in its fields.
4. Click **"Create and issue pass"** (or click **"Create"**, select the newly created person in the list, and click **"Issue pass to selected person"**).
5. In the pass issuance dialog:

- add the card: present it at any reader or enter the card code manually;
- specify the start date and end date for the pass validity period;
- in the **"Access time"** field, select **"Unlimited"** or another required schedule mode;
- in the **"Access Objects"** section, select the object (for example, `Entrance A`).

6. Verify the entered details and click **"Save"**.

Result: the person is issued a pass with the assigned identifier and access permission for the `Entrance A` object.

Verification: the newly created pass appears in the passes list with the status "Active", the configured validity period, and the `Entrance A` access object.

7.5. Testing Access and Verifying Journal Events

Before testing, verify that the `Entrance A` object is in the "Configured" state, communication with its equipment is "Active", and the issued pass is currently valid.

1. Open the "Access Objects" workspace and select the `Entrance A` object.
2. In the bottom "Events" panel, enable filtering by the selected object.
3. Present the issued card at the reader.
4. Verify the hardware response: when access is granted, the lock of the access object should unlock according to the object's timing settings.
5. Verify that a new record appears in the "Events" panel:

- result — "Access granted";
- reason — "Granted by pass";
- participants — the access object, person, pass, and identifier.

6. Open the "Journals" workspace and locate the same record in the main event feed (category "Access").

Result: the system recognized the identifier, granted access to `Entrance A`, and recorded the event in the journal.

To test an access denial, present an unregistered card at the reader: an "Access denied" event should appear in the journal with the reason "Access denied: identifier not found".

If no access event appears: check, in order, the communication state of the line and peripheral device, the reader binding on the access object, whether the pass is active, and whether the raw card read signal is visible in **"Commissioning (LIVE)"** mode.

Troubleshooting recommendations are provided in Chapter 16.

7.6. Further System Configuration

Initial configuration of Seenaptec Access Platform (SAP) is now complete: equipment is connected, access objects are configured, passes are created and issued, and access events are being recorded in the event journal.

You can continue configuring the system according to your site requirements:

- access schedules, conditions, perimeters, and APB rules — see Chapter 11;
- access templates, pass requests, and custom person fields — see Chapter 12;
- operator accounts, roles, and permissions — see Chapter 14;

- HTTPS setup, backup and restore, and system maintenance — see Chapter 10.

Chapter 8. UI Basics

All workspaces in the Seenaptec Access Platform (SAP) web interface follow a single layout pattern. This chapter describes the common elements and interaction principles; the “System Overview” start page is covered in Chapter 9, and each individual workspace is described in Chapters 10–14.

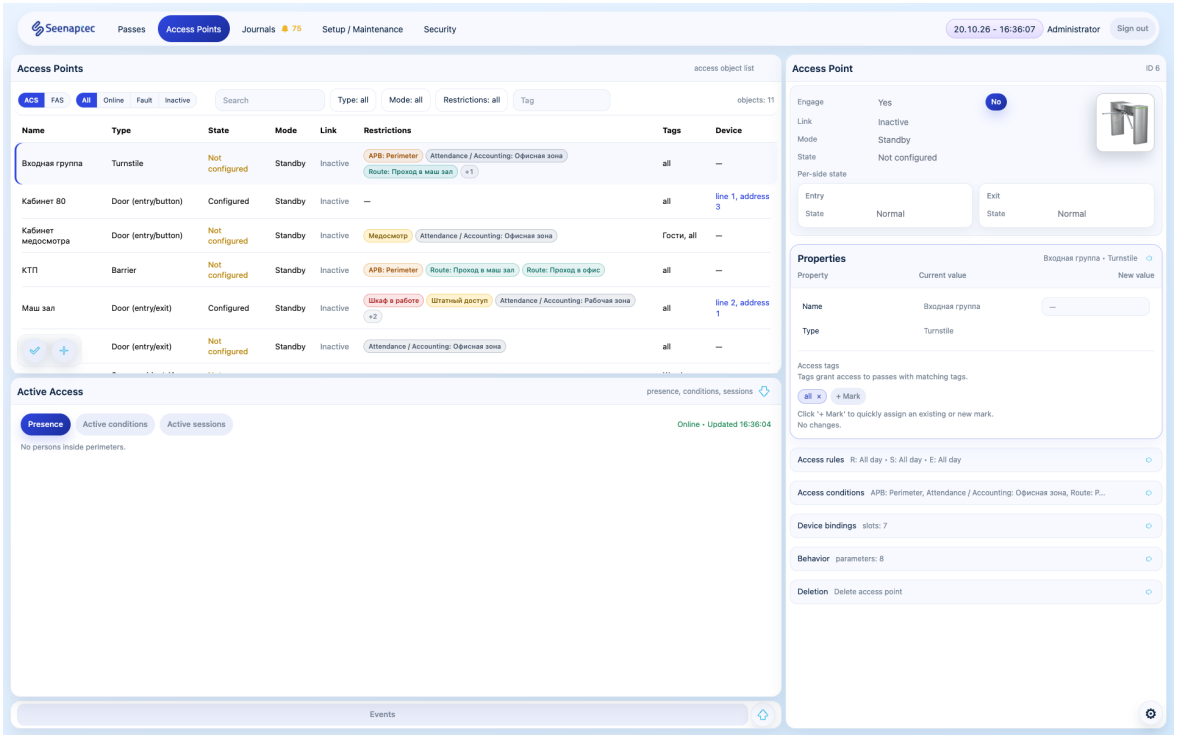
8.1. Signing In

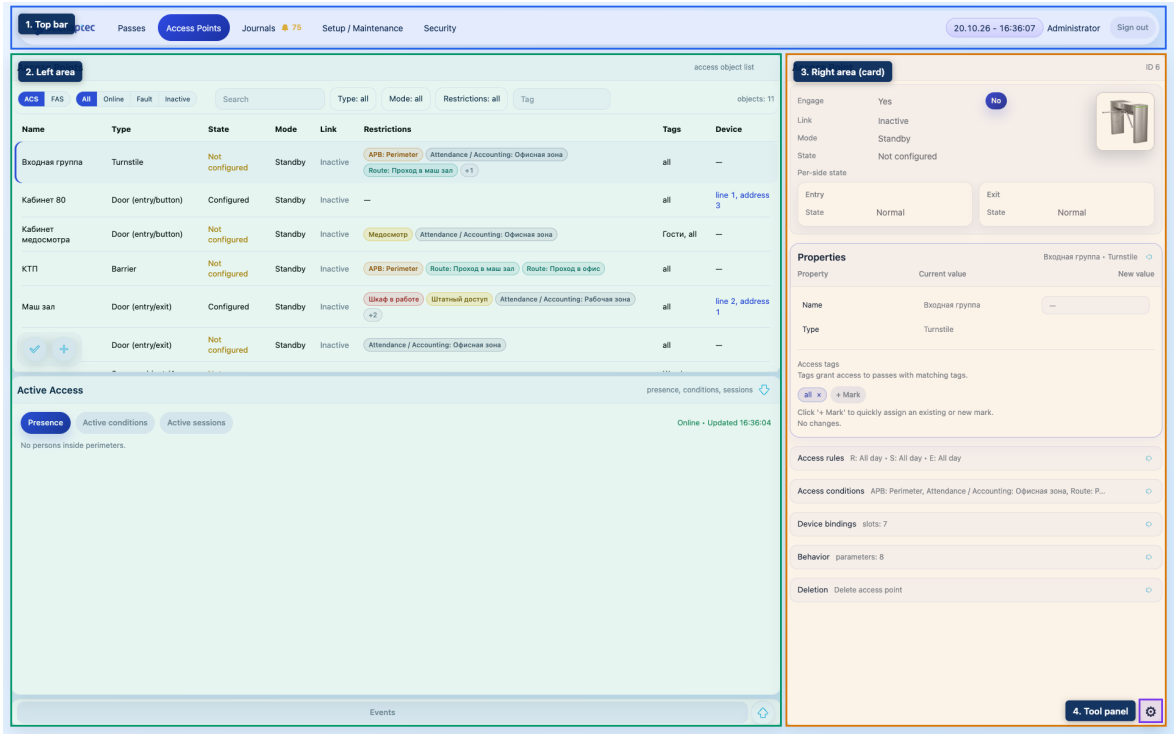
The web interface opens at `http://<server-address>:8080/app/`. Before starting work, the system prompts for the operator’s username and password (the “Sign in to SKD” screen). On the very first launch, while the system is not yet activated, the “System activation” screen is displayed instead of the sign-in screen to enter the activation key (see Chapters 6 and 17).

The set of available workspaces, panels, and actions depends on the operator’s role: for example, a pass office operator may not see the “Setup / Maintenance” workspace. In this and subsequent chapters, the interface is described in full—as seen by an administrator.

The current operator’s name is displayed on the right side of the top bar, alongside the “Sign out” button, which ends the session.

8.2. Workspace Screen Layout





The workspace screen consists of four main areas:

1. **Top bar** — contains buttons for switching between system workspaces, an indicator showing the presence or absence of alarm messages, the current date and time, the operator's name, and the sign-out button.
2. **Left area** — contains panels with lists, object trees, and event feeds. Depending on the selected workspace, either a single panel or multiple vertically stacked panels may be displayed.
3. **Right area** — contains the card of the selected system entity, such as a line, device, access object, person, or pass.
4. **Tool panel** — slides out from the right edge of the screen and contains directories, parameters, and additional operations for the selected workspace.

When bulk operation mode is enabled, a panel for selecting and configuring an action to be applied to all selected list items appears on the right.

The composition and contents of the areas depend on the selected workspace and the operator's permissions.

To resize the width of the left and right areas, drag the vertical splitter between them. If two panels are open in the left area, adjust their height using the horizontal splitter.

The chosen layout is saved in the browser and restored the next time the operator opens the web interface.

8.3. Top Bar and Workspaces

The top bar contains five workspace buttons:

Workspace	Purpose	Left-Area Panels
"Passes"	Persons, identifiers, pass issuance	"Passes", "Events", "Identifiers"
"Access Objects"	Doors, cabinets, rules, and active access (`ACS`) / fire zones `D`, `A`, `P` (`FAS`)	"Access Objects" ("FAS Zones"), "Active Access", "Events"
"Journals"	Historical event analysis, alarms, reports	"Journals", "Alarms"
"Setup / Maintenance"	ACS and FAS equipment, commissioning, system operations	"Equipment", "Events", "Debug"
"Security"	Operators, roles, sessions, secrets, platform audit	"Operators", "Events", "Sessions"

The active workspace is highlighted. To the left of the workspace names is the system logo—clicking it opens the "System

Overview" start page (Chapter 9); to the right are the **ACS | FAS** subsystem switcher, the current date and time, the operator's name, and the "Sign out" button.

Subsystem switcher (ACS | FAS). On the right side of the top bar, a compact pill switches the active subsystem (when both subsystems are enabled on the server and the operator's role has access to both):

- in the **"Setup / Maintenance"** workspace, the pill switches the left "Equipment" tree between the **ACS** domain (controller, RS-485 lines, and OSDP peripheral devices) and the **FAS** domain (UniPOS fire panels, system blocks, signaling loops, and addressable detectors/modules; see Section 10.9), while simultaneously switching the context of the lower "Events" panel and the live trace source selector (`srv_osdp` or `srv_ops`) in the "Debug" panel;
- in the **"Access Objects"** workspace (whose top-bar button is labeled **"Objects"** when the FAS subsystem is enabled), the pill switches the workspace between ACS access objects and FAS fire zones (`D`, `A`, `P`; see Section 11.12), automatically filtering the lower "Events" panel by the active subsystem;
- in the **"Journals"** workspace, the pill expands into a three-state selector **All | ACS | FAS**, acting as a quick subsystem filter for events and alarms (Chapter 13).

If only one subsystem is enabled in the installation or the operator's role is restricted to a single subsystem (Section 14.3), the `ACS | FAS` switcher is hidden automatically.

Alarm indicator. If there are active unacknowledged alarms in the system, a yellow bell icon with an alarm counter is displayed on the "Journals" workspace button. The arrival of a new alarm is accompanied by a sound alert. Click the icon from any section of the system to open the "Alarms" panel of the "Journals" workspace. When there are no active alarms, the icon is hidden.

8.4. Left Area: Panels

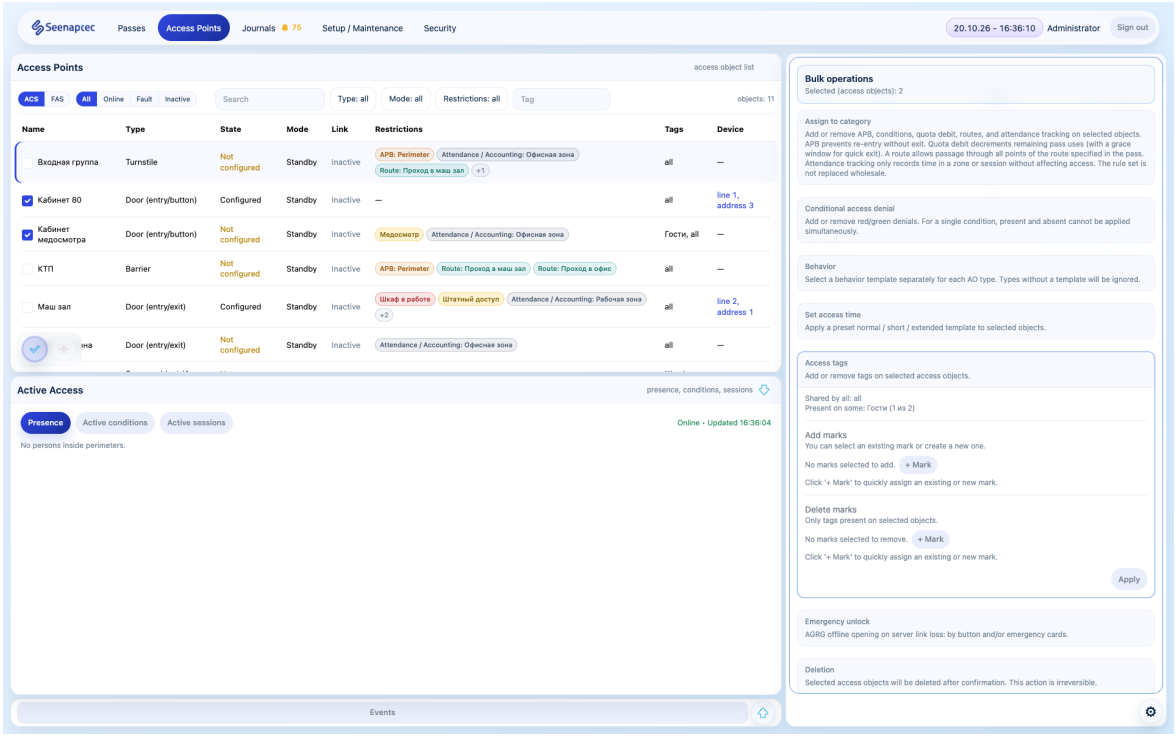
Each workspace can have several left panels (see the table above). The first panel is the primary one and is always open. Any other panel can be opened **beneath it**—pinned to the second slot: collapsed panels are shown as buttons at the bottom of the left area, and clicking a panel title or its pin button expands the panel. An open lower panel can be collapsed using the unpin button in its header or by clicking empty space in the header. The selected combination of panels is remembered separately for each workspace.

Panels follow a uniform structure:

- **list or tree** — rows with key columns and color-coded status indicators (for example, green "Active", red "Offline"); clicking a row selects the entity and opens its card in the right area;
- **action buttons** — a group of buttons for list operations is located in the bottom-left corner of the panel.

Creating entities is done with the "+" button in the bottom bar of the panel: it opens a creation dialog—for example, "Add line", "Add access object", "Add person", or "Add pass". The dialog requests the minimum required fields; the remaining settings are then configured in the card. Next to "+" are controls for enabling bulk operation mode and workspace-specific actions (for example, "SSB scan on current line" or "Network adapter configuration" in the "Equipment" panel).

Enabling bulk operation mode is done with the "V" button in the bottom bar of the panel: it opens the bulk operation selection panel on the right.



Deleting entities is not performed directly from the list: single deletion is done from the entity card (the “Deletion” / “Delete …” section or button at the bottom of the card), while bulk deletion is performed in the bulk operations panel in selection mode.

Destructive operations always require confirmation listing the affected records.

8.5. Right Area: Card

The card displays the selected entity and is divided into **sections**—thematic blocks with headers: for example, a pass card has “Information”, “Identification”, and “Pass”; a device card has model details, components, and settings. Sections collapse and expand when their header is clicked; an arrow next to the header indicates the section state. The set of open sections is remembered per workstation separately for each card type (person, device, journal event, etc.) and applies to all records of that type across page reloads. The composition of sections depends on the entity type and is described in Chapters 10–14.

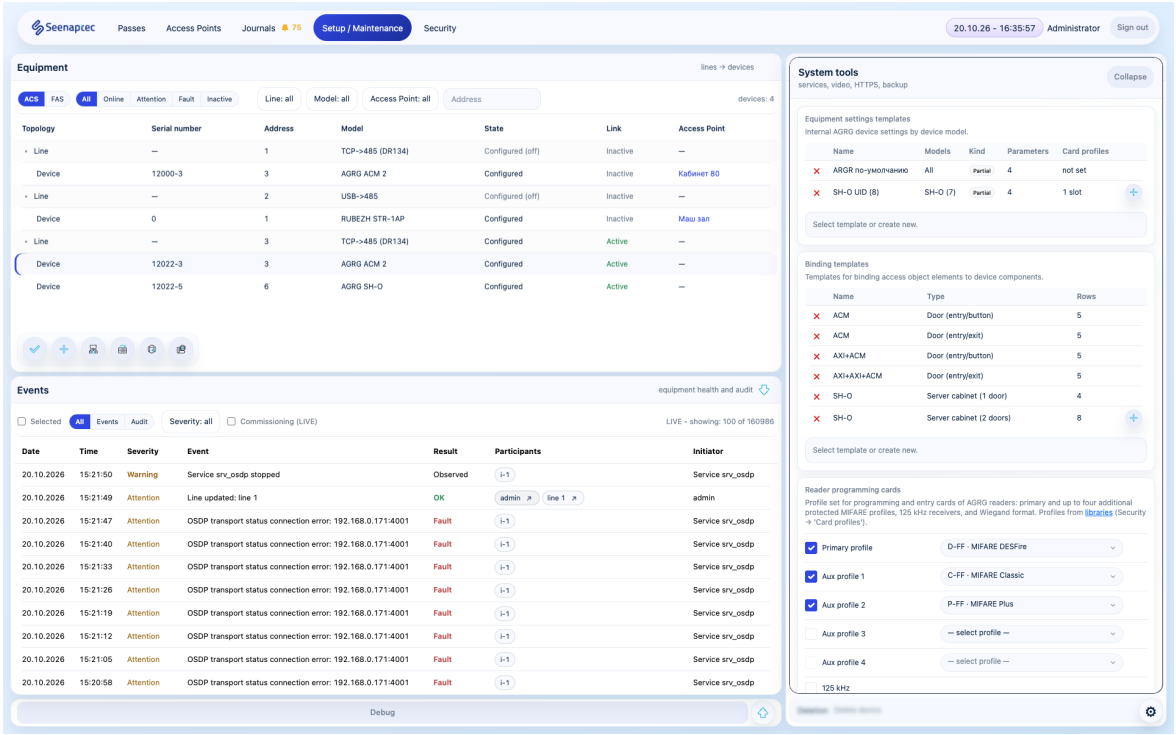
The card operates in a “view + edit” model:

- in its normal state, a section displays current values—you can safely read the card without changing anything;
- in most cases, a new value is entered into a separate field so that both the previous and new values are visible; modifying any field puts the section into a dirty state: “Save” and “Reset” buttons appear, and new values can be entered for multiple fields at once; changes are not written to the server until saved;
- for reference lists (templates, roles, rules), the editor is opened with an explicit “Edit” button and works the same way.

Saving is confirmed by a notification, and the change is recorded in the audit log with a list of “before/after” field values.

Buttons with a yellow background indicate sending commands to hardware.

8.6. Tool Panel



To open the tool panel, click the gear button near the bottom-right corner of the screen. The panel contains directories and settings for the active workspace. The available fields and sections in the tool panel depend on the operator's permissions.

Panel contents by workspace:

- **"System Overview"** — "Language", "Appearance", "Subsystems", "Support", "Access recovery", "Attention thresholds" (Section 9.6);
- **"Passes"** — "Access templates", "Identifier templates", "Custom personnel fields", "Auto-cleanup";
- **"Access Objects"** — "Access rules", "Behavior templates", "Access time template", "Production calendar";
- **"Journals"** — "Report generation", "Archiving";
- **"Setup / Maintenance"** — system tools: equipment settings templates and binding templates, "Device firmware", "Backup / restore", "HTTPS access", "Service Management";
- **"Security"** — "Roles", "Secrets", "Security settings".

Modifying parameters in an open tool panel follows the standard directory workflow: select a record, make changes, and click **"Save"**. To discard unsaved changes, click **"Reset"**.

To close the tool panel:

- click the **"Collapse"** button on the panel;
- click the gear button near the bottom-right corner of the screen again;
- click any panel in the left area.

Switching to any other workspace collapses the tool panel automatically.

8.7. "Events" Panel

The "Events" panel displays both newly arriving and previously recorded events related to the active workspace. New records appear automatically without refreshing the page.

The scope of events depends on the selected workspace and the active `ACS | FAS` subsystem toggle:

- **"Access Objects"** — in `ACS` mode, displays access events and changes to access objects; in `FAS` mode, displays events for fire zones (`D`, `A`, `P`), alarms, activations, bypasses, and operator actions on zones;
- **"Passes"** — person access events, as well as changes to persons, identifiers, and passes;

- **“Setup / Maintenance”** — in `ACS` mode, displays line and OSDP device state changes; in `FAS` mode, displays connectivity, composition discovery, fault, and command events for UniPOS panels, loops, and addressable devices;
- **“Security”** — platform audit events.

Working with the event feed:

- use filters by time period, severity, event type, and text;
- enable the **“Selected”** toggle to show only events for the entity currently selected in the primary panel—for example, the history of a specific access object or pass;
- click an event to open its card showing the date and time, initiator, result, reason, and “before/after” value changes;
- click a participant badge to filter events by the associated access object, device, pass, person, or identifier;
- click the arrow on a participant badge to open that entity’s card. When navigating to another workspace, a back button appears in the panel header to return to the previous view;
- click **“Open in Journals”** to jump to the selected event in the “Journals” workspace.

Active filters are displayed above the feed. To remove a participant filter, click its chip.

Each event record preserves the participant labels that were in effect at the time of registration. As a result, the event remains intelligible even after the associated entity is deleted.

In the “Events” panel of the “Setup / Maintenance” workspace, an additional **“Commissioning (LIVE)”** mode is available. This mode displays low-level hardware signals that are not stored in the journal. For details, see Section 10.6.

8.8. Verifying Command Execution — “Command Accepted Does Not Mean Completed”

Some operations execute asynchronously. When you click a button, the system first accepts the command and then forwards it to the appropriate service or hardware device. Therefore, a message confirming that a command was accepted does not confirm that the operation itself has finished.

Asynchronous operations include, for example:

- unlocking or opening a door;
- restarting a service;
- scanning for devices;
- updating device firmware.

Verify command completion using one or more of the following indicators:

- **state change** — the card and list reflect the new state of the object, for example a line transitions to “Active”;
- **outcome event** — an entry appears in the “Events” panel reporting successful command execution or an error;
- **progress indicator** — for a long-running operation, its current execution progress and final result are displayed.

Do not treat an operation as completed based solely on the command acceptance notification.

If the operation state has not changed and no completion or error event has appeared after the system accepted the command, check connectivity between the server, the line, and the field device. Do not resend the command until you have verified the operation status. For details on equipment diagnostics, see Chapter 10.

8.9. General UI Behavior Rules

- **Live updates.** Lists, states, and feeds update automatically. This also applies to the operator’s own actions: saved changes are immediately reflected across all panels where the modified record participates—for example, issuing an identifier in a person’s card is immediately visible both in the “Identifiers” panel and in the passes list. If the browser loses connection to the server, data freshness is not guaranteed; the live server date and time in the top bar serve as an indicator of an active connection.
- **Unsaved changes.** If you attempt to navigate away from unsaved changes (by selecting another record or closing an editor), the interface asks for confirmation before discarding them.
- **Confirmations in the right area.** Action confirmations appear over the right area with cancel and proceed buttons; destructive actions are highlighted in color.
- **Operator permissions.** Buttons and sections unavailable to the operator’s role are hidden or disabled with an explanation. Permissions are described in Chapter 14.

- **Unified vocabulary.** All UI labels use the terminology defined in Chapter 3; states and decision reasons are displayed using identical wording both in workspace panels and in the journal.

8.10. Practical Tips for Working with the Interface

Related entities can be opened directly from lists, cards, and events, including across workspaces.

Main navigation shortcuts:

- the arrow on a participant badge in the “Events” panel or the “Journals” workspace opens the associated person, pass, identifier, access object, device, line, or server;
- a value in the “Device” column of the access objects list opens the device in the “Setup / Maintenance” workspace;
- a value in the “Access Point” (“Access Object”) column of the equipment tree opens the object in the “Access Objects” workspace;
- the **“Open in Journals”** button opens the selected event in the “Journals” workspace.

After a cross-workspace jump, a back button appears on the left side of the primary panel header with the tooltip **“Return to … with previous selection”**. It returns you to the originating workspace and restores:

- the selected record;
- the open card;
- the set of open panels;
- the selected event in the journal or event feed.

The system remembers only the most recent jump. The back button disappears if you manually switch workspaces in the top bar or navigate via the alarm indicator.

Performing operations from different sections

Several operations are accessible from multiple places in the interface. Choose the workflow that best fits your current task:

- you can bind equipment to an access object from the access object card, from the device card, or via a bulk operation;
- you can add an identifier from the “Identifiers” panel or from a person’s card;
- you can issue a pass from the “Passes” panel, from a pass request, or based on an access template.

Regardless of the method chosen, the result of the operation is saved identically in the system and recorded in the audit log.

Chapter 9. System Overview

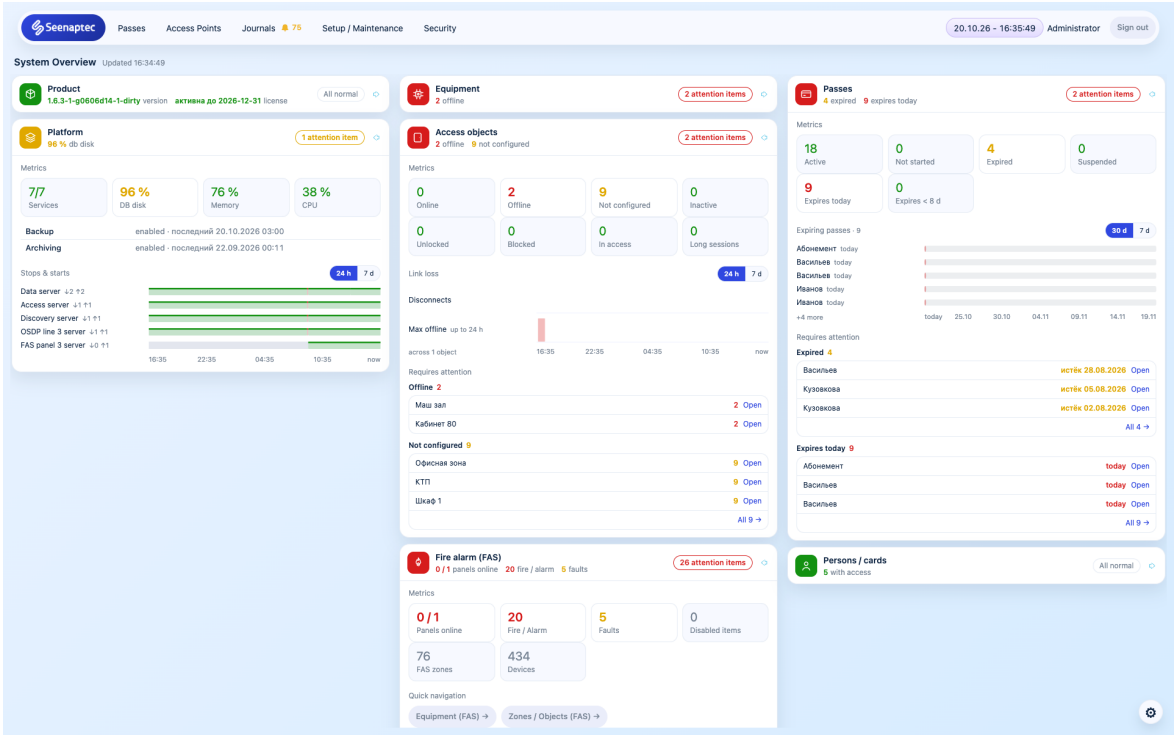
“System Overview” is the start page of the web interface: a single-screen summary of the entire system’s health. The page answers two questions—“is everything normal?” and “what requires attention right now?”—and provides direct links to the source of any issue. The workspaces targeted by these links are described in Chapters 10–14.

9.1. How to Open the Page

- Click the system logo in the top-left corner of the screen. Clicking it again returns you to the workspace from which “System Overview” was opened. On narrow screens where the top bar collapses into a menu, “System Overview” is the first item in that menu.
- The page opens automatically after signing in, as well as on the very first launch of the interface on a workstation.
- Reloading the browser page restores the last open workspace: if “System Overview” was open before the reload, it will open again.

The page is accessible to all roles; the set of status cards does not depend on the operator’s permissions. Individual sections of the tool panel (Section 9.6) are displayed according to permissions.

9.2. Status Cards



The screen displays status cards across the key domains of the system:

Card	What It Evaluates
Product	software version and build, server ID, license expiration
Platform	server services, database disk usage, host memory and CPU, backup and journal archiving
Equipment (`Hardware`)	connectivity with devices and line link quality
Access Objects	connectivity with access objects, abnormal states (unlocked objects and long sessions)
Fire safety	connectivity with FAS panels, standby mode, fire alarms, faults, and bypasses of FAS zones and devices (displayed when the FAS subsystem is enabled)
Passes	active and problematic passes, expiring validity periods
Persons/cards	persons and identifiers: blocks, expired cards, access denials

The cards are arranged in one to three columns depending on the window width.

Card status is indicated by the color of the icon and badge next to the title:

- **green** — normal;
- **yellow** — warning: the system is operational, but there are items worth reviewing;
- **red** — intervention required;
- **gray** — no data: the metric cannot be calculated (for example, no lines are configured); does not affect the overall status.

A card's status is the worst status among its metrics. Next to the title is a badge showing either "N attention items" or "All normal".

Collapsing. Cards with issues are expanded by default, while healthy cards are collapsed. Clicking a card header expands or collapses it; your manual choice is preserved while the page remains open (automatic refreshes do not reset it). Reloading the page resets expansion states according to current health.

9.3. Metrics and Additional Sections

Inside an expanded card is the "Metrics" section: rows of "label — value", with the value colored according to its status. When issues are present, the "Requires attention" section appears below (Section 9.4).

"Product" — software version and distribution, brand name and color scheme, developer and copyright holder, build info (date, commit, Go version), and server ID. The "License" metric monitors the expiration date: it turns yellow when fewer than 30 days remain, and red when fewer than 7 days remain (thresholds are configurable, see Section 9.6). For the license lifecycle, see Chapter 17.

"Platform" — "Services" (how many system services are running out of the total), "DB disk", "Memory", "CPU", "Backup" (age of the latest backup), and journal "Archiving". The additional "Stops & starts" section displays a timeline chart of service activity over the last 24 hours or 7 days (such as **Application server**, **Behavior server**, **Device discovery server**, **OSDP line N server**, and **FAS panel N server**).

"Equipment" — device summary: "Online", "Offline", "Replacement", "Disabled". The "Link quality" metric lists up to the three most problematic lines along with their specific condition:

Line Condition	Status	Meaning
"disabled"	yellow	the line is administratively disabled
"port unavailable"	red	the line service is running, but the transport port failed to open
"offline"	red	the port is open, but there is no communication with devices
"possible line break"	yellow	polling is active, but the bus has been silent for a minute or longer—a line break is likely (Section 10.1)
"N% errors"	yellow / red	share of erroneous replies on the line: yellow from 5%, red from 20%

The additional "Link loss" section shows an hourly chart of devices going offline over the last 24 hours or 7 days.

"Access Objects" — summary of access objects: "Online", "Offline", "Not configured", "Inactive"; abnormal states: "Unlocked", "Blocked", "In access", "Long sessions" (an object has been held open longer than the threshold—60 minutes by default). Its own "Link loss" section breaks down connectivity losses by access objects.

"Fire safety" (FAS subsystem) — summary of UniPOS fire panels and zones: how many panels are online ("Panels online" / "Offline"), whether the subsystem is in normal standby mode, and how many active fire alarms/pre-alarms ("Fire / Alarms"), technical faults ("Faults"), and disables/bypasses ("Disabled / Bypass") are active across zones (`D`, `A`, `P`), loops, and addressable devices. When a fire alarm, panel communication loss, or fault occurs, the card expands automatically and lists the affected panels, zones, and detectors under "Requires attention" with one-click navigation to "Access Objects" (`FAS` mode) or "Setup / Maintenance" (`FAS` mode).

"Passes" — "Active", "Not started", "Expired", "Suspended", "Expires today", "Expiring < N d" (configurable threshold, 7 days by default). The additional "Expiring passes" section lists the next five passes expiring within a 30-day or 7-day window.

"Persons/cards" — persons: total, with access, blocked, "on site"; cards: total, free, expired, blocked and lost. The additional "Access denials" section shows unknown cards and the persons with the highest number of access denials over the last 24 hours or 7 days.

Additional sections with charts are loaded when a card is expanded and when the time period is switched; they are not part of the general periodic page refresh.

9.4. "Requires attention" List

This section appears inside a card when it has problematic metrics. For each such metric, the first three affected objects are listed along with their specific condition—for example, "Line 5 (possible line break)". If there are more than three objects, the list ends with an "All N →" link.

- **“Open”** next to an object — navigates to that object in the corresponding workspace: to the line row in “Setup / Maintenance”, or to the card of the access object, pass, or person.
- **“All N →”** — navigates to the full filtered list of problematic objects in the target workspace.

When navigating from the overview, the target workspace remembers the return point: the back button in its panel header takes you back to “System Overview”.

If there are no issues, “All normal” is displayed instead of the list.

9.5. Automatic Refresh

- Page data refreshes automatically every 60 seconds; the timestamp of the latest update is shown in the page header (“Updated …”).
- Automatic refresh requests are background service requests: they validate the session but do not extend it. Once the inactivity timeout expires (Section 14.6), signing in again will be required even if the page was refreshing the entire time.
- Additionally, every 30 seconds the browser sends a background `/health` request so that the connection through the platform proxy server does not go idle; this does not affect operator activity or session lifetime.
- If data fails to load, the page displays an error message and a “Retry” button.

9.6. Tool Panel

The tool panel opens via the gear button near the bottom-right corner of the screen (general operation is described in Section 8.6). It includes the following sections:

“Language” — interface language selector (`Russian` / `English`). The selected language is saved in the operator’s browser and immediately switches all interface labels and event journal dictionaries without reloading the page.

“Appearance” — interface theme (“Light” / “Dark”) and page background for the dark theme (“Graphite”, “Indigo”, “Teal”, “None”). This setting is stored in the server configuration and applies to all roles and the entire web interface, including the sign-in screen. Changes take effect after clicking “Save”; “Reset” restores the default theme and background.

“Subsystems” — installation-wide activation of platform functional domains: **“Access Control System (ACS)”** and **“Fire Alarm System (FAS)”** (at least one subsystem must remain enabled). When both subsystems are enabled, the `ACS | FAS` switcher appears in the top bar and status cards for both subsystems are shown on the overview page.

“Support” — installation identifiers for contacting technical support: Server ID, installation code (fingerprint), and serial number. The “Copy all” button copies all three values as a single block suitable for pasting into an email or support ticket; the “Fingerprint composition” block displays the core hardware devices from which the installation code is derived. How the fingerprint works, when it is generated, and when it changes are described in Section 17.7.

“Access recovery” — configuration for signing in with a one-time vendor code if the administrator password is lost. This section is visible to operators with permissions to view or manage operators; modifying the settings requires full operator management permission. For details, see Section 17.8.

“Attention thresholds” — thresholds at which overview metrics turn yellow (labeled “amber” in field names) and red:

Field	Default
“License expiring (amber)” / “… (red)”	30 / 7 days
“Passes expiring”	7 days
“Long unlock session”	60 min
“DB disk usage (amber)” / “… (red)”	80 / 90 %
“Memory usage (amber)”	85 %
“CPU usage (amber)”	90 %

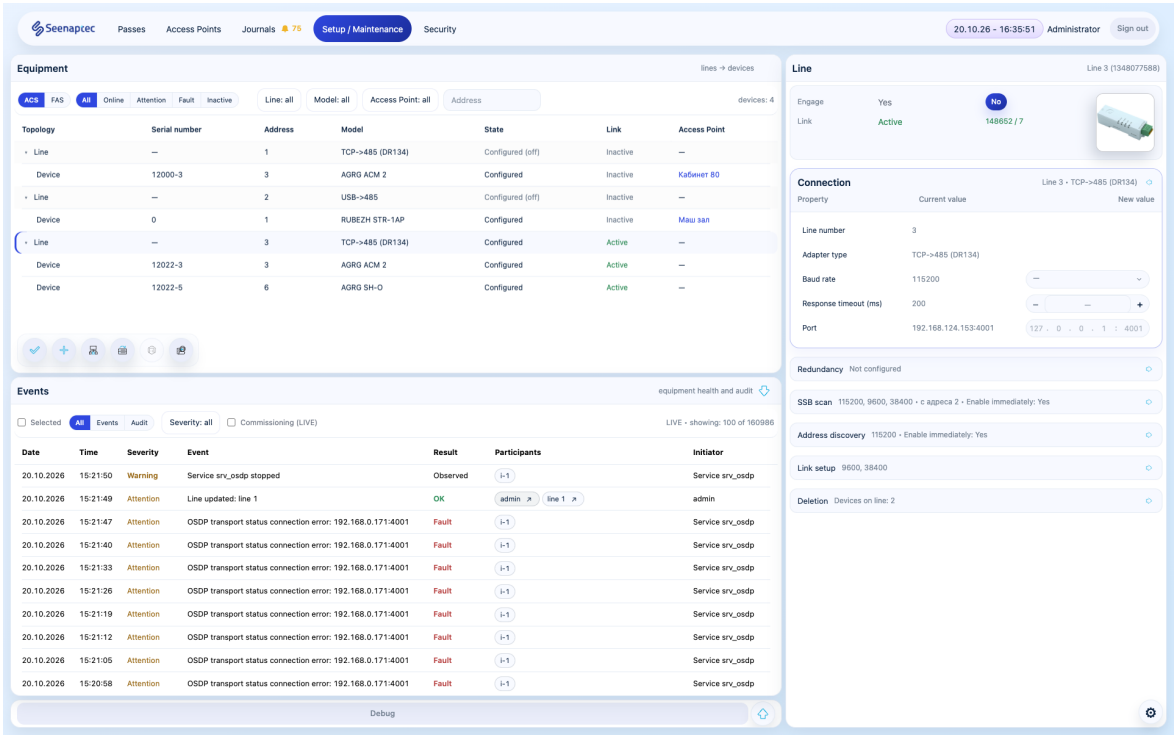
Thresholds are system-wide: they are stored in the server configuration and apply to all roles. After saving, metrics are recalculated immediately without reloading the page. Line link quality thresholds (5% and 20% errors) are fixed and not exposed in the panel.

Chapter 10. Setup / Maintenance

The “Setup / Maintenance” workspace is the engineer’s workbench: connecting and servicing hardware, commissioning, diagnostics, and system operations. Its left panels are “Equipment”, “Events”, and “Debug” (with the top subsystem switcher **ACS | FAS** when fire alarm integration is enabled); the tool panel houses system utilities: templates, firmware, backup and restore, HTTPS, and service management.

10.1. “Equipment” Panel

The panel displays the “lines → devices” equipment tree. The list columns are: “Topology”, “Serial number”, “Address”, “Model”, “State”, “Link”, and “Access Point” (“Access Object”).



- For a **line**, the “Address” column shows the line number, “Model” shows the adapter type, and “State” shows “Configured” or “Configured (off)”.
- For a **device**, the row shows the serial number, OSDP address, vendor and model, lifecycle state, link state, and the names of bound access objects; clicking the “Access Point” column opens the object in the “Access Objects” workspace (if multiple objects are bound, the first one is opened), and you can return using the back button in the panel header (Section 8.10).

Device lifecycle states: “New” (discovered by a scan, awaiting confirmation), “Modified” (the device reported a revision—a change in its model details or component composition), “Configured” (saved in the system configuration), “Unknown” (no physical communication has occurred yet, device details have not been received); the “(off)” suffix indicates that the hardware is currently manually disabled.

Link states: “Active” (communication is running; highlighted in blue when operating over Secure Channel), “Offline”, “Possible break” (lines only: the adapter is working and polling is active, but no device has responded for over a minute—a bus break or power loss to all devices is likely; shown in yellow), “Inactive” (the line is disabled), “Not serviced” (the line service is stopped), “Suspended” (for example, during a scan), “Port unavailable” (no communication with the adapter). For a redundant ring line, the row additionally displays a ring mode badge: “Split: segmented polling”, “Primary adapter active”, or “Secondary adapter active” (when one loop leg fails, the badge names the active adapter), or “Line down” (under normal conditions when the ring is intact, no badge is shown). The link state of a ring line is determined by whether polling is occurring: as long as communication runs through at least one port (including segmented polling during a bus break or operation via the secondary adapter), the line and its devices remain “Active”, while the failure of a specific port is visible on its indicator in the line card; “Port unavailable” is shown only when both ports are unavailable.

Above the list are quick status filters (**All | Online | Warning | Fault | Inactive**) and filters by line, model, state, link, access object, and address; on the right is the “devices: N” counter.

Panel action buttons (bottom left):

- “Device selection mode” — enables multi-selection for bulk operations (Section 10.5);
- “Add” — opens a menu for “Line” / “Device” (a device is added to the currently selected line);
- “Network adapter configuration” — discovery and configuration of network RS-485 converters (Section 10.3);
- “SSB scan on current line” (Section 10.3);
- “Bulk create/bind access objects” — batch creation of access objects from devices according to project data (by serial numbers) (Chapter 11).

10.2. Lines

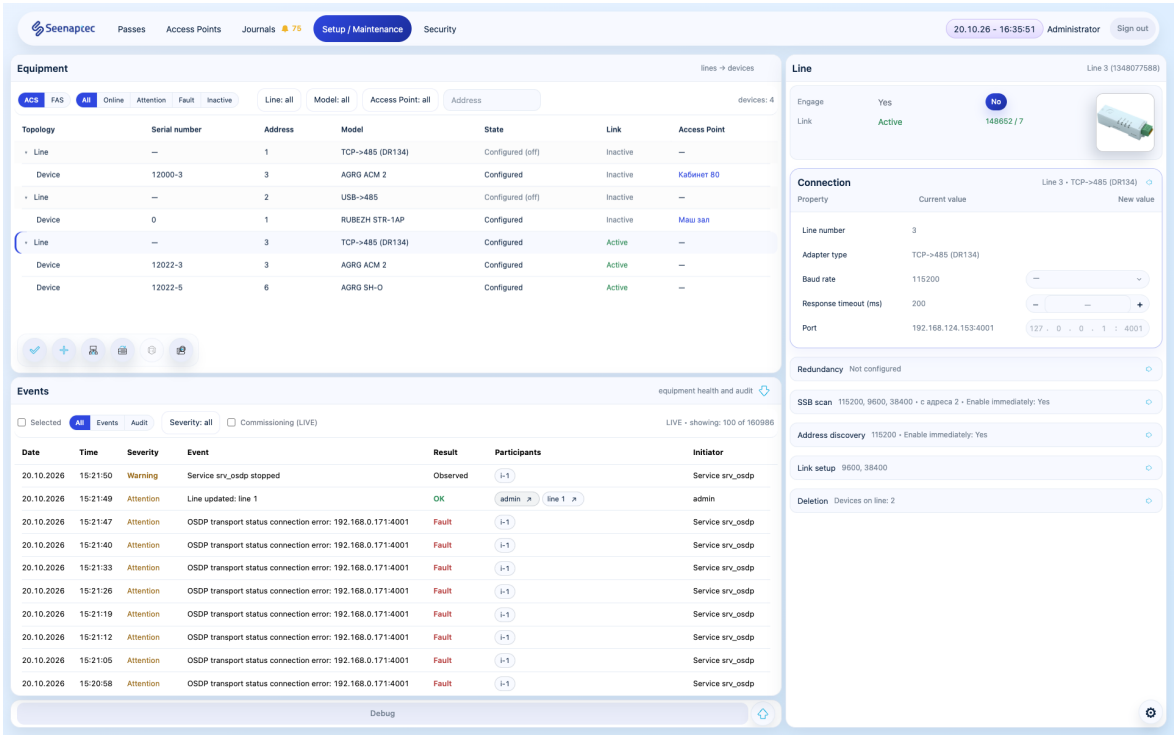
Creating a Line

Clicking “Add” → “Line” opens the “Add line” dialog:

- “Adapter type” — `USB->485`, `TCP->485 (Other)`, `TCP->485 (DR134)` (adapter characteristics are described in Chapter 5);
- “Primary port” — serial port name (for example, `/dev/ttyUSB0`) or `host:port` address for TCP;
- “Secondary port” — if specified, redundancy is enabled automatically;
- “Line number”;
- “Baud rate” — for network converters that negotiate baud rate automatically, “Not applicable” is shown.

A new line is created in the enabled state. After creation, the `srv_osdp` service for the line starts automatically; readiness is indicated by the “Active” link state.

Line Card



At the top of the card are the “Engage” toggle for manual disabling, the current link state with communication statistics, and—when redundancy is enabled—the “Ring” row: the ring mode badge and the “Primary” / “Secondary” port indicators (connection, receive activity, echo). Indicator dots: “connection” — the port is open; “receive” — data was recently received from the port; “echo” — frames sent through this port were recently heard at the opposite end of the ring. When a port fails, all of its dots turn off immediately; after port recovery, “echo” lights up only upon the first confirmed frame.

Card sections:

- **“Connection”** — line number and adapter type (read-only), baud rate, reply wait time (“Response timeout (ms)” field), and port.
- **“Redundancy”** — enable toggle and secondary port. The primary port is inherited from the line settings; failover parameters do not require manual tuning—timeouts and migration delays are built into the system.
- **“SSB scan”** and **“Address discovery”** — launch device discovery (Section 10.3).
- **“Link setup”** — switch devices to the line’s baud rate (for hardware that supports baud rate changes).
- **“Deletion”** — a line can be deleted only when no devices are attached to it.

There are no start/stop buttons for the line itself: a line is either engaged or disengaged; the services backing it are managed separately (Section 10.8).

Redundancy

Redundancy protects a single RS-485 bus segment accessible via two communication paths (separate converters and cable routes) connected as a ring. Both paths are served by a single `srv_osdp` line instance: under normal conditions, all devices are polled through the primary port while the secondary port listens to the bus and monitors ring integrity by checking echoes of transmitted frames. If the bus breaks (“Split: segmented polling”), the devices in each segment are automatically routed to whichever port can reach them; if the primary path fails completely (converter failure or TCP connection loss), polling switches to the secondary port (“Secondary adapter active”). Once a segment recovers, devices return to the primary port automatically after a hold-off delay to prevent flapping. For day-to-day operation, the summary states in the list and card are sufficient: the ring mode badge, port indicators, and link state; any device currently polled via the secondary port displays a “Secondary” chip in the list and on its card. Breaks, recoveries, and polling route switches are recorded as technical state events.

10.3. Discovering and Adding Devices

SSB Scan

For supported AGRG OSDP devices, automated Seenaptec SecureBus discovery is used: discovery by serial number, address assignment, and baud rate negotiation. The “SSB scan on current line” button opens a dialog with a table of discovered devices (address, serial number, vendor/model, baud rate, status). Before applying, you can select the connection mode (including switching to Secure Channel) and an equipment settings template by device type; clicking “Apply selection” admits the checked devices into the configuration. If not all expected devices appear, run the scan again.

SSB scanning can also be launched from the “SSB scan” section of the line card, where you can specify the baud rates to sweep, the starting address (“Address from”), and the “Enable immediately” option.

Address Discovery

For third-party OSDP equipment, use the “Address discovery” section of the line card: it sweeps a range of addresses (“Start address” — “End address”) across the selected baud rates. Discovered devices appear in the results block and can be added to the configuration.

Manual Addition

Clicking “Add” → “Device” opens the “Add devices to line” dialog: enter OSDP addresses as a comma-separated list or range (for example, `1–8, 10, 12–15`)—the dialog shows how many records will be created and which addresses are already occupied. Created records are enabled immediately; connectivity appears once the physical devices respond to polling.

Network Adapter Configuration

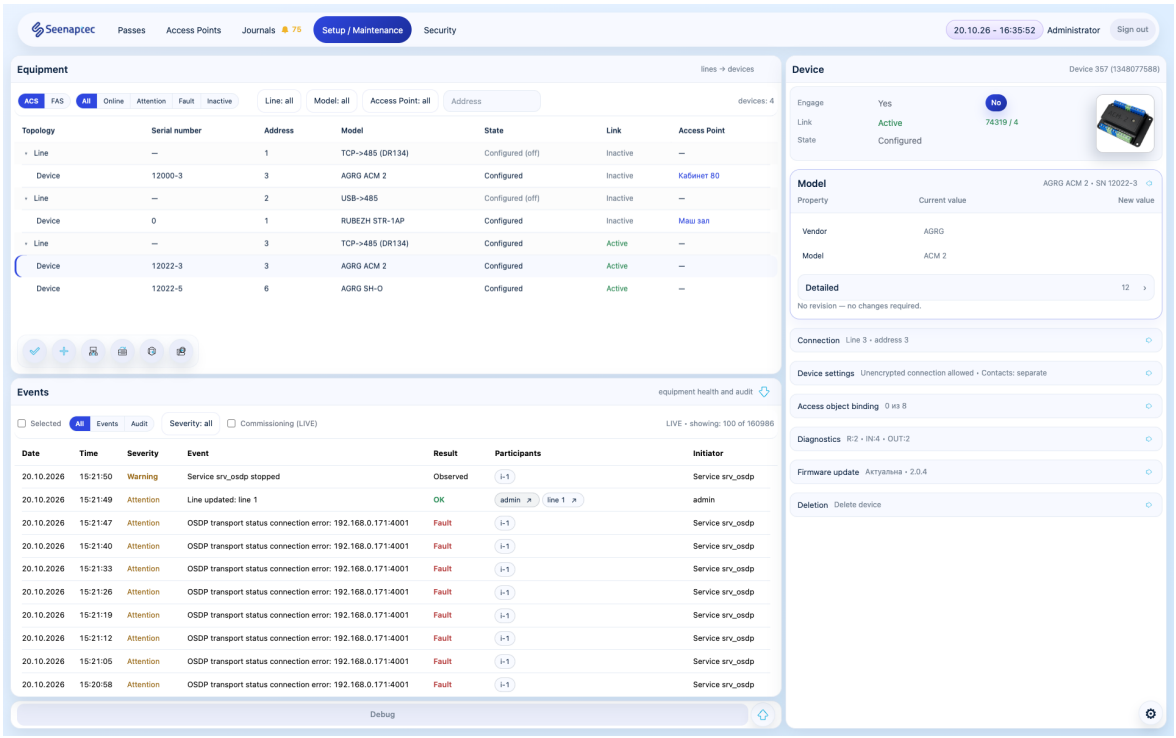
The “Network adapter configuration” button opens the discovery and setup tool for network RS-485 converters (currently only the DR134 model is supported) on the local network: scanning, verifying and assigning IP addresses, importing “serial number → IP” mappings from CSV, and creating lines with primary and secondary loop roles. Adapters with an IP address in the server’s current subnet are configured normally; discovered adapters with an IP address from a different subnet can also be configured—commands are delivered directly by MAC address (on macOS/Linux servers), and such rows are marked with an “other subnet” informational badge in the list. For an adapter already added to the system, changing its IP address remains available, for example to migrate it into the current subnet.

10.4. Devices

Accepting a New Device

A device found by discovery is saved in the “New” state and remains inactive until confirmed. The card of a new device displays its line, OSDP address, discovery mode, serial number, and model; clicking “Acknowledge” (“Accept”) brings the device into the active configuration.

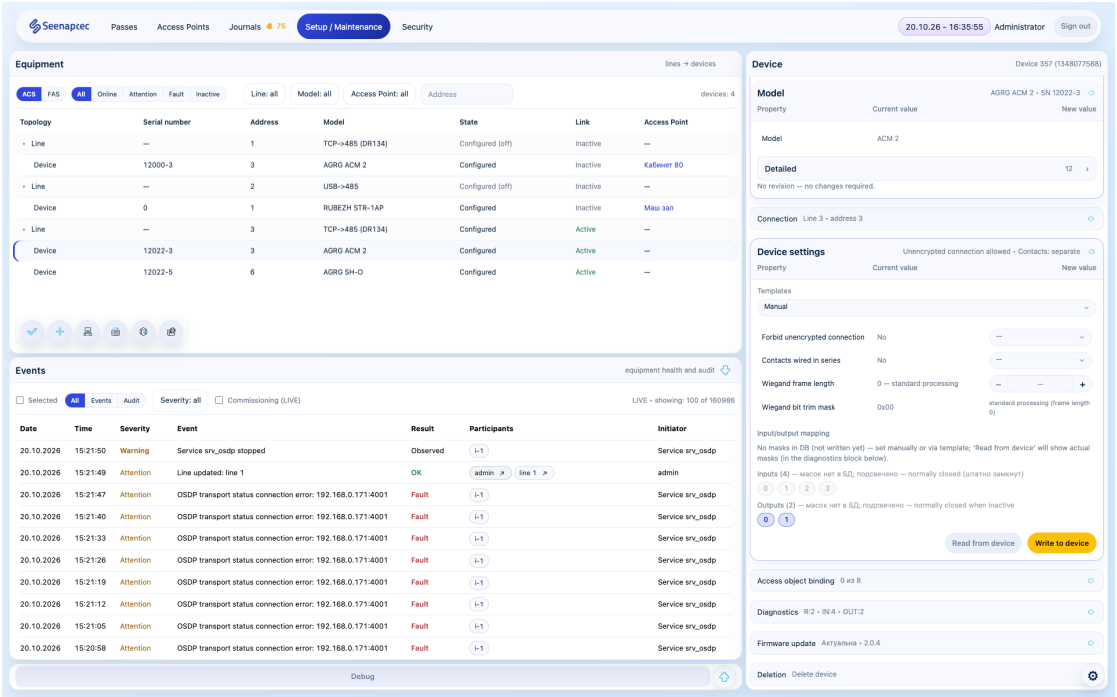
Device Card



At the top of the card are the “Engage” toggle, link state, lifecycle state, and an image of the device. Sections:

- **“Model”** — device identity: vendor and model; when expanded, it shows hardware version, serial number, firmware version, and the list of device capabilities. If the device reports a **revision**—a change in identity or component composition—the section offers “Accept changes” or “Reset”; if the changes would break existing bindings, confirmation is performed via the “Accept and break bindings” button, which then navigates to the bindings section.
- **“Connection”** — line and OSDP address on the line (including changing the address by sending a command to the device), as well as OSDP Secure Channel management. Modes: “Unencrypted”, “Compatible (SCBK-D)”, “Initial key installation”, “Fully encrypted”, and “Fully encrypted with automatic key” (automatic key rotation). The section displays the active and pending keys, encryption status, and next rotation time, and provides a confirmed device key reset action.
- **“Device settings”** — AGRG hardware parameters. This section operates on devices with firmware version 2.0.0 and above: on earlier versions, a placeholder prompts you to update the firmware, and until the firmware version is polled, it prompts you to wait for the device identity poll. Values are stored in the system database and shown in the “Current value” column; edits are made in the “New value” column. The **“Write to device”** button sends the settings to the device and saves them to the database upon success; **“Reset”** discards unsaved edits. The **“Read from device”** button performs a diagnostic read: actual hardware values are displayed under a “Read from device (diagnostics)” banner with “matches DB” / “differs from DB” indicators—reading does not save anything or overwrite current edits. Parameters include: “Forbid unencrypted connection”, “Contacts wired in series”, and Wiegand transport bit trimming—“Wiegand version” (frame length in bits; `0` disables trimming) and “Wiegand bit trim mask” (which frame bits are stripped when forming the card code, such as parity bits). This section also includes the **“Input/output mapping”** editor: the polarity of each input (“normally open” / “normally closed”) and the idle state of each output are configured point by point using numbered point buttons (zero-based numbering: input 0, output 0, etc.—consistent with the rest of the system). Input polarity on the device combines with the slot’s “Inversion” flag in the access object’s behavior settings (Section 11.3):

enabling inversion at both levels cancels out. For SH-O readers, this section also configures the card reading chain—the **"Card profiles"** block: an ordered list of protected card profiles from the library (Section 14.9) that the reader tries from top to bottom, plus UID reading. Settings can be saved as a template and applied to devices of the same model (Section 10.8). A device may reject parameters as inapplicable in its current state: in that case, the operation fails immediately with a "Parameters not applicable" error and an explanation rather than waiting for a response timeout. For example, "Forbid unencrypted connection" cannot be enabled until an encryption key is installed on the device—install the key in the "Connection" section first.



- **"Access object binding"** — mapping of device components to access object slots (Chapter 11); the summary shows "N of M" bound components.
- **"Diagnostics"** — live state and direct control of components: readers (last read code, LED and buzzer control), contacts (tamper, power, inputs), relays (0.5–2 s pulse, turn on/off). Direct control is unavailable when the device is offline.
- **"Firmware update"** — compares the device's reported firmware version against files in the firmware catalog and writes firmware to the device (Section 10.8) through the stages: "Preparing", "Downloading image", "Verifying checksum", "Transferring to device", "Activating", "Verifying version after write", and "Completed".
- **"Deletion"** — removes the device from the configuration; associated access objects are preserved, and their bindings to this device are cleared. For AGRG devices with firmware version 1.x, a "Factory reset" button is also available here: after confirmation in the dialog, the device is reset to factory defaults and removed from the system—use this when decommissioning hardware from a site. The reset is confirmed by the device itself: if it does not respond or rejects the command, deletion is not performed.

Commands to devices execute asynchronously (Chapter 8): always confirm the outcome via device state and events.

Replacing a Device

Bindings to access objects are linked to the device record in the configuration rather than to a specific physical unit, so you do not need to delete the device record when replacing faulty hardware. Mount the replacement device on the same line with the same OSDP address; if the new device responds at a different address, update the address in the existing device record ("Connection" section → "Save"—this updates the configuration record without sending a command to the hardware).

When different hardware starts responding at the record's address, the server receives identity details that differ from the stored ones and registers a **revision**: the device transitions to the "Modified" state, and the "Model" section displays the detected changes. If the replacement device is of the same model, accepting the revision ("Accept changes") updates the identity—serial number and firmware version—while preserving all access object bindings with no further action required. If the replacement is

a different model and the change in component composition breaks existing bindings, confirm via “Accept and break bindings” and then adjust the bindings in the access object card (using a binding template or manually, see Chapter 11).

If the device record was operating over Secure Channel, a key must be installed on the replacement unit—check the encryption status in the “Connection” section.

10.5. Bulk Operations

The “Device selection mode” button enables multi-selection in the equipment list; the right area is replaced by the “Bulk operations” panel showing the count of selected devices. Only one section is active at a time:

Section	Operation
“Connection mode”	Enable/disable devices and switch Secure Channel mode (including key installation) for a group of devices
“Create/bind access object”	Create and bind access objects for the selected devices; binding is performed via a binding template (selected automatically) or manually
“Equipment settings templates”	Apply settings templates grouped by “vendor/model”
“Firmware”	Batch firmware upload (the matching firmware file is selected from the catalog automatically)
“Deletion”	Delete the selected devices

Execution progress is shown by indicators in the “State” column of the list (“Selected”, “Pending”, “Applying”, and the final outcome for each operation).

10.6. Equipment Events and “Commissioning (LIVE)” Mode

The “Events” panel displays:

- line and device state changes;
- service starts and stops;
- equipment setting changes showing previous and new values;
- operation outcomes and error details.

Filters, the **“Selected”** toggle, and navigation via participant badges are described in Section 8.7.

“Commissioning (LIVE)” Mode

The **“Commissioning (LIVE)”** mode is designed for verifying incoming hardware signals during installation and troubleshooting.

To enable this mode, check the **“Commissioning (LIVE)”** box in the “Events” panel. The feed will additionally display low-level signals, including:

- identifier reads;
- keypad presses;
- input or output state changes;
- Secure Channel state changes;
- device firmware update stages.

Such entries are prefixed with **“LIVE •”**, appear only while the mode is enabled, and are not saved to the persistent journal.

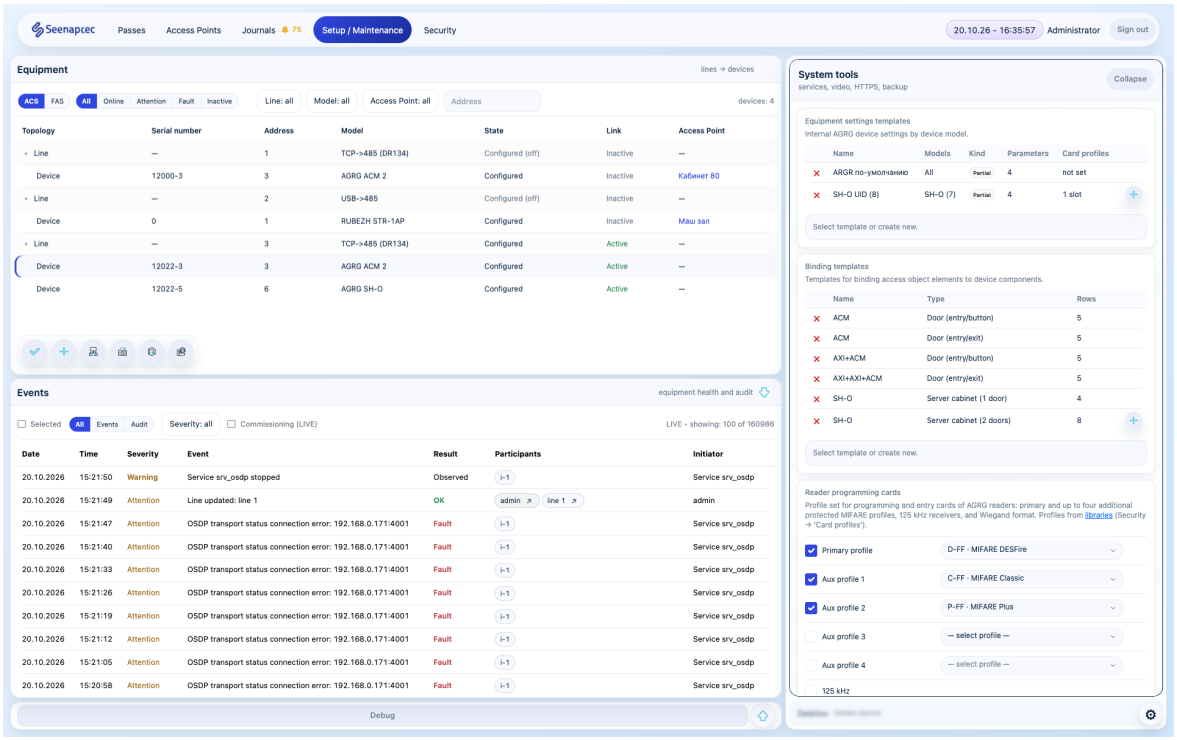
To test a reader, enable LIVE mode, present a card, and verify that an entry with the read identifier code appears in the feed. The appearance of the entry confirms that the signal reached the server, but does not imply that access was granted or that a command was executed by the hardware.

Disable **“Commissioning (LIVE)”** mode after completing your checks.

10.7. “Debug” Panel

The “Debug” panel shows runtime logs from the service instance handling the selected hardware node: in `ACS` mode, the `srv_osdp` instance for the selected line or device; in `FAS` mode, the `srv_ops` instance for the selected fire panel (Section 10.9). It provides substring search, a log level filter (`EMERG`…`DEBUG`), log clearing, and export to a file. A redundant ring line is served by a single `srv_osdp` instance, so its log is unified—events from both ring ports appear in a single stream. Log publishing is enabled via the “MQTT” toggle; when publishing is turned off, the panel indicates this. The panel is intended for in-depth troubleshooting; for routine operation, states and events are sufficient. Logs can be exported to a file and shared with technical support.

10.8. System Tools (Tool Panel)



“Equipment settings templates”

Named sets of AGRG device settings: parameters from the “Device settings” section (including Wiegand transport bit trimming), input/output mapping, and—for templates targeting SH-O readers specifically—the card profile chain. A template can be created from an existing device’s settings or directly in the section editor, and can be applied during an SSB scan, from a device card, or in bulk operations. Input/output mapping is stored point by point in the template, matching the “Input/output mapping” editor in the device card. Applying a template requires firmware version 2.0.0 or higher: a device running an earlier version rejects the operation with a prompt to update firmware.

“Binding templates”

Preconfigured “device component → access object slot” mappings for standard wiring schemes—the recommended way to bind hardware to access objects. They are used in the access object card (“Binding mode” selector), when creating objects from selected devices, and during bulk creation (Chapter 11); a template can also be saved from a manually assembled binding.

“Reader programming cards”

Commissioning AGRG readers in the field using physical cards: an entry card puts a factory-new or reset reader into programming mode for 30 seconds, and within that window a programming card writes the site’s card profiles into the reader. This section prepares both card types; an already commissioned reader enters programming mode via an entry card matching any profile in its active set. Working with this section requires the “Equipment: Reader programming cards” permission scope;

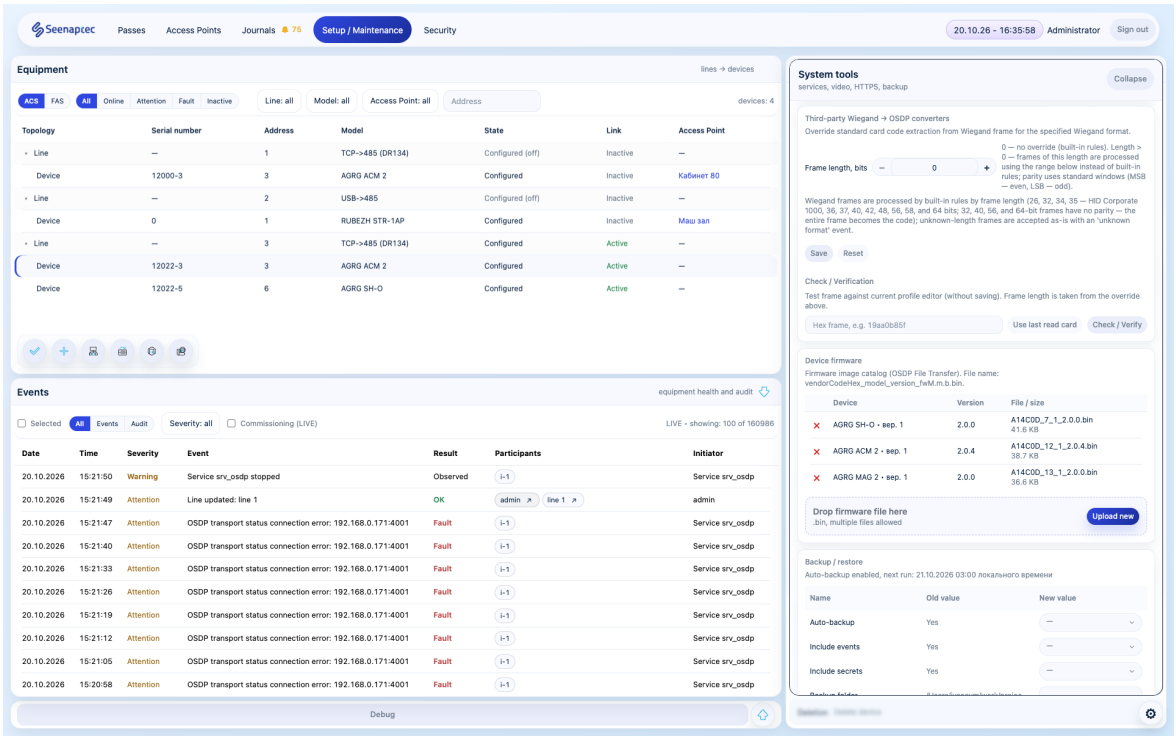
card issuance is performed on a desktop reader assigned to the operator's workstation (Section 14.8) with the "Programming cards" permission enabled on its card.

In this section, you configure a unified profile set: a primary protected MIFARE profile, up to four additional profiles, and 125 kHz card receivers—profiles are selected from the library (Section 14.9); a common Wiegand output format (26, 34, or 58 bits with standard parity bits) is chosen for the set. Clicking **"Create card"** saves the set and opens the issuance dialog; clicking "Reset" reverts the set to its last saved state. In the dialog, choose the issuance target ("Target profile (product type)" field):

- **"New reader entry card"** — for a factory-default or reset reader;
- **"Entry card for profile …"** — for an already configured reader, matching one of the profiles in the set;
- **"Programming card"** — carries the profile set to the reader; presented within the programming mode window after the entry card.

Issuance is a single-pass operation: the dialog automatically prepares the credential medium (transitioning a blank or previously used card to the required key state: factory keys for a new reader entry card, the selected profile's keys for a profile entry card, or the manufacturer key for a programming card) and immediately writes the payload. Keep the card on the desktop reader until completion; progress is shown in the dialog log. Data is written starting at block 4 (sector 1); sector 0 of the card is not modified. Every issuance is recorded in the audit log.

"Third-party Wiegand → OSDP converters"



Overrides the default extraction of card codes from Wiegand frames—used for third-party Wiegand-to-OSDP converters through which third-party readers are connected to an OSDP line. By default, the system applies built-in rules for standard frame lengths; configure an override when the frame format is non-standard:

- **"Frame length, bits"** — length of the incoming frame; `0` disables the override (built-in rules apply), valid values are 2–64;
- **bit range** — which frame bits form the card code; bits are numbered from zero starting on the right (least significant bit); by default, the entire frame is used;
- **bit tape** — a visual diagram of the frame: most significant bit on the left, parity bits marked with "P", code bits highlighted, and stripped bits dimmed; you can select the range by dragging across the tape.

A live test tool is provided below: enter a raw frame in hexadecimal or populate the last read frame using the "Use last read card" button—the system will show the resulting card code under the current editor settings. You do not need to save the profile to test it.

Saving a modified profile requires confirmation: the new profile changes how incoming frames are interpreted, so card codes enrolled under the previous profile will no longer match new reads. Stored card codes are not recalculated automatically—migrating to a new format requires re-issuing identifiers (Section 12.6) or re-importing them with the appropriate interpretation specified (Section 12.9).

“Device firmware”

Server-side catalog of firmware files: upload files (via drag-and-drop or the “Upload new” button) and delete them. The file name encodes the vendor, model, and version—the system uses this metadata to automatically match the appropriate firmware file to a device. Writing firmware to hardware is performed from the device card or via a bulk operation.

“Backup / restore”

Backup and recovery of system configuration and data:

- automatic backup settings: “Auto-backup”, “Include events” (include the journal in the backup), “Include secrets” (include secrets in the backup), “Backup folder”, “Retain files”, and “Start time”;
- “Password” — the backup is encrypted using a selected secret from the “Security” secret store; the “Unencrypted” toggle creates a plaintext backup file (for isolated test benches). The password is never displayed in the UI;
- “Include secrets” — includes all system secrets (including the license activation key) and the secret store master key in the backup. Such a backup is always encrypted, so this option is available only after a “Password” secret has been selected. A backup file containing secrets is marked with “contains secrets” in the backup list—protect it just like a password;
- “Create backup now” — triggers an immediate manual backup;
- recovery: select a backup file (“Select backup”). For an encrypted backup, use the “Backup password” block: if a backup password secret is configured in settings, it is substituted automatically; otherwise, enter the password manually. The same block is used when opening an encrypted journal archive (Chapter 13). Clicking “Recover” starts the restore process; afterward, system services restart automatically and the panel displays their status until all services are ready. When restoring a backup that contains secrets, the system displays an additional warning that current secrets and the master key will be overwritten.

License-related recovery rules (see Section 17.6 for details):

- you can restore only a backup created on **the same installation**: a backup taken from a different installation (with a different license server ID) is rejected;
- restoring onto a clean, not-yet-activated system is permitted—this is how migration to a new server is performed; after restoration, the system will prompt you to re-enter the activation key for that installation. The exception is a backup created with “Include secrets” enabled: the activation key is restored from the backup, and re-entering it is not required;
- the runtime database of active presence sessions and access object sessions is not included in backups. After restoration, active sessions are cleared except for hardware-initiated states (such as forced-open / tamper alarms).

Backup creation and restoration are recorded in the journal; a backup failure generates a critical-severity event.

“HTTPS access”

Enables native HTTPS without an external reverse proxy: “HTTPS” toggle, port (`8443` by default), and TLS certificate. Upload a certificate into the secret store using the “Add HTTPS certificate to store” button and select it from the list; the status shows “HTTPS saved, pending apply” and then “HTTPS accepting connections”. Changes to HTTPS configuration are recorded in the audit log.

“Service Management”

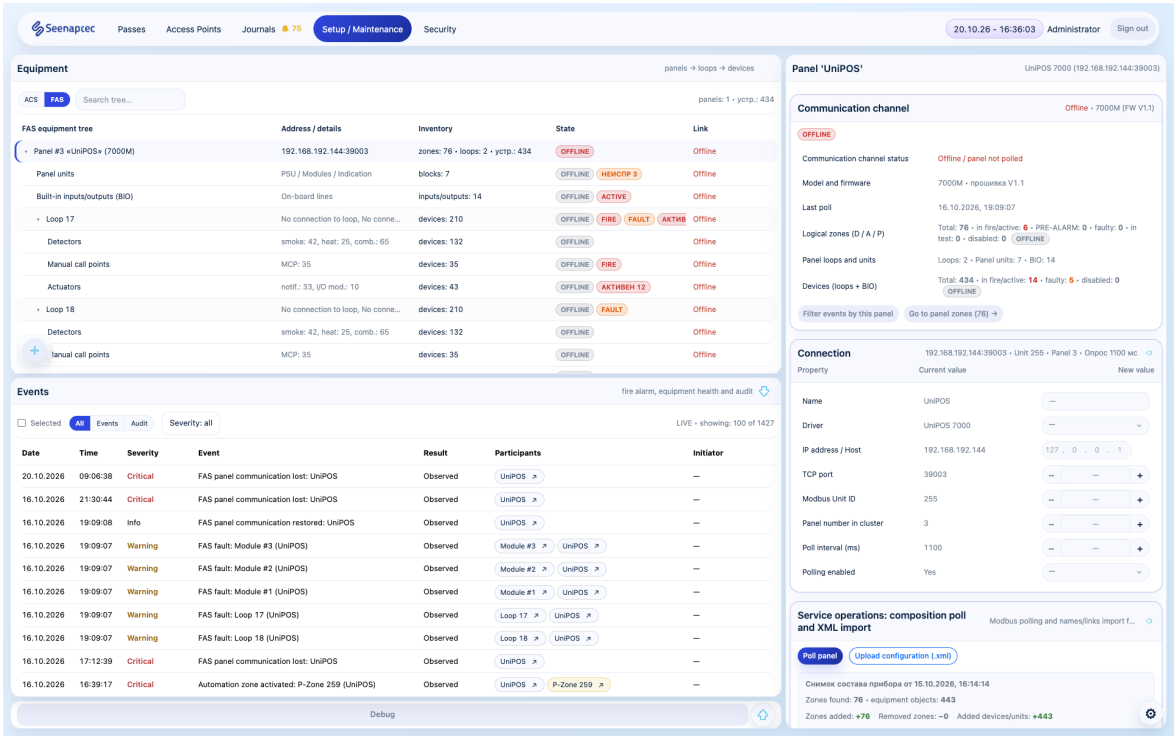
List of platform services with their current states and “Start / Run”, “Stop”, and “Restart” commands: “Data server” (**Application server**), “Access server” (**Behavior server**), **Device discovery server**, “OSDP line server N” (**OSDP line N server**, including redundant instances), and **FAS panel N server**. Commands are asynchronous: the operator request is recorded in the audit log, while actual service startup and shutdown are recorded as technical state events.

The tool panel also contains the “Video server” section, where the video server address is configured and the camera list is loaded. Until a video server address is configured, video integration is considered inactive and all video-related UI elements (the “Monitoring” section and camera bindings in access object cards) are hidden; the “Video server” section itself remains

accessible. Video subsystem integration is not covered in detail in the current edition of this guide and will be documented in upcoming software releases.

10.9. FAS Equipment (UniPOS Panels)

To configure and diagnose fire alarm and automation equipment, switch the **`ACS | FAS`** pill on the right side of the top bar to **`FAS`**. The “Setup / Maintenance” workspace switches to the FAS hardware domain: the left “Equipment” panel displays the tree of UniPOS IFS7000 series fire panels, while the lower “Events” and “Debug” panels switch to the event stream and live trace of the `srv_ops` service.



FAS Equipment Tree and Connectivity Indication

In `FAS` mode, the left “Equipment” panel organizes the physical hierarchy of fire panels with the columns `Name`, `Type`, `Address`, and `State`:

1. **FAS Panel** — the root node representing a UniPOS IFS7000 control panel (`7000M`, `7000-1MC`, `7000M-1L`). To register a new panel, click the **“+”** button (“Add FAS panel”) in the bottom bar of the “Equipment” panel and enter its name, IP address, TCP port (`502` by default), and cluster panel number (`1`–`8`).
2. **System blocks and signaling loops** — child nodes of the panel created automatically after a composition scan:
   - power supply unit (`PSU`);
   - indication and notification blocks (`BIO`) and relay/input expansion blocks (`BRV`);
   - ring signaling loops (`Loop 1`, `Loop 2`, etc.).
3. **Addressable FAS Devices** — fire detectors (optical smoke, thermal, combined, gas, manual call points `MCP`), input/output modules, and automation control devices (`FPE`), grouped inside loops and `BIO`/relay blocks by functional role: **“Detectors”**, **“Notification”**, **“Automation control (FPE)”**, and **“Inputs & outputs”**.

Connectivity and state indication in the tree:

- for a panel and its signaling loops, active communication with the hardware is highlighted in green as **“Online”** / **“Active”**;
- for system blocks and addressable devices, the normal standby state is shown in plain text as **“Normal”** without visual clutter, whereas abnormal conditions are color-coded: **“Fire”** / **“Pre-alarm”** (red), **“Fault”** / **“Short circuit”** / **“Line break”** (amber), **“Test”**, and **“Disabled (bypass)”**;

- if TCP connectivity with a panel is lost, both the panel itself and **all of its child blocks, loops, and addressable devices** automatically transition to **"OFFLINE"** (their last known states prior to communication loss are retained only as a reference note inside the card and never mask the offline status).

FAS Panel Card

Selecting a FAS panel opens its card in the right area, organized into four sections:

1. **"Communication channel"** — a non-collapsible summary at the top of the card. It displays the current link status (`Online`, `Offline`, `Disabled`), network address `IP:port` and `Unit ID`, panel number in the cluster (`1`–`8`), device identification passport (model `IFS7000`, serial number, and firmware version read via Modbus `0x2B / 0x0E`), timestamp of the last successful poll, and a composition summary (counts of blocks, loops, addressable devices, and zones).
2. **"Connection"** — a collapsible configuration table in `Property | Value | New value` format:
   - `Name` — display name of the panel;
   - `Polling enabled` — enables or administratively disables polling of the panel by `srv_ops`;
   - `IP address` and `TCP port` — 4-octet IPv4 address input and Modbus TCP port (`502` by default);
   - `Modbus Unit ID` — Modbus unit identifier (`255` by default for UniPOS IFS7000 panels);
   - `Cluster panel number` — panel number in the UniPOS cluster network (`1`–`8`);
   - `Poll interval, ms` and `Response timeout, ms` — cyclic register polling interval (`1000` ms by default) and reply timeout (`2500` ms by default). After entering new values, click **"Save"** to apply the changes (or **"Reset"** to discard edits).
3. **"Service operations"** — panel maintenance commands:
   - **"Scan composition"** — triggers a Modbus TCP inventory scan of the panel: the service queries system block slots, signaling loops, addressable devices, and zones (`D`, `A`, `P`) and populates the equipment tree while preserving existing UUIDs and custom names;
   - **"Reset alarms and faults"** — sends a command to the panel to reset latched fire alarm and fault states;
   - **"Upload XML configuration"** — imports a project `.xml` file exported from the official UniPOS configuration software. During import, the system automatically populates custom names for fire zones (`D`, `A`, `P`), loops, `BIO` outputs, and all addressable devices, and binds detectors and modules to their respective zones.
4. **"Delete"** — deletes the FAS panel and its associated object tree after confirmation in the right area.

Cards of System Blocks, Loops, and Addressable Devices

- **System block cards (`PSU`, `BIO`, relay blocks) and loop cards (`Loop N`)** display a node status summary and a non-collapsible **"Composition"** section listing all sub-blocks, channels, and addressable devices belonging to that block or loop along with their addresses, types, bound zones, and current states. Clicking any device row in the "Composition" table opens that device's card.
- **Addressable FAS device cards** display the detector or module passport (category, subtype—such as optical smoke, rate-of-rise thermal, manual call point, relay module, or `FPE` device—loop number and address on the loop, bound FAS zone, and detailed state/contamination/measurement-error flags) and a **"Control"** section with operator commands:
  - **"Enable (Put in service)"** — returns the device from bypass or test mode to normal standby operation;
  - **"Test"** — places the device into service test mode;
  - **"Disable (bypass)"** — temporarily excludes a faulty or serviced detector/module from generating alarms.

Modbus TCP Diagnostics in the "Debug" Panel

When a FAS panel (or any of its loops or devices) is selected, the lower **"Debug"** panel attaches to the live diagnostic stream of the corresponding `srv_ops` instance ("FAS panel N server"). With the **"MQTT"** toggle enabled and the log level set to `DEBUG`, the feed displays a real-time **Modbus TCP (MBAP)** frame trace:

- **`TX`** — outgoing requests showing the function code, `Unit ID`, transaction ID (`tx`), starting register address (`addr`), register count (`qty`), and full packet HEX dump;
- **`RX`** — successful panel replies showing request duration (`dur`), payload byte count, and HEX dump;
- **`RX EXC`** — Modbus exception replies (such as `Illegal Data Address`) with the exception code, duration, and HEX dump;
- **`ERR`** — socket errors, read timeouts, or MBAP header mismatches.

The trace covers all four Modbus function codes used by the driver: device identification passport reads **`0x2B / 0x0E`** (`Read Device Identification`), holding and input register reads **`0x03`** (`Read Holding Registers`) and **`0x04`** (`Read Input Reg-`

isters), and panel text archive reads **0x14** (Read File Record). You can filter the trace by substring (for example, 0x14, RX EXC, or ERR) and export it to a text file for analysis.

Chapter 11. Access Objects

The "Access Objects" workspace is the access administrator's workbench: logical access points, their access rules, and a real-time view of "who is where". Its left panels are "Access Objects", "Active Access", and "Events"; the tool panel contains "Access rules", "Behavior templates", "Access time template", "Production calendar", "Routes", and "Presence tracking".

11.1. Types of Access Objects

An access object is a logical entity for which permissions, rules, and operating modes are configured. Its **type** defines its entire behavior in the system: which hardware slots the object has and which operating modes and rules apply to it (for example, anti-passback perimeters for doors and turnstiles, or access sessions for server cabinets). Hardware is bound to slots separately and can be bound later: an access object can be created before physical installation, and field devices can be replaced without losing permissions or event history (Section 10.4). Permissions are always assigned to the access object rather than to a controller (as in legacy ACS platforms) or reader, and all access decisions are made by the server rather than by a peripheral controller board.

This approach allows you to:

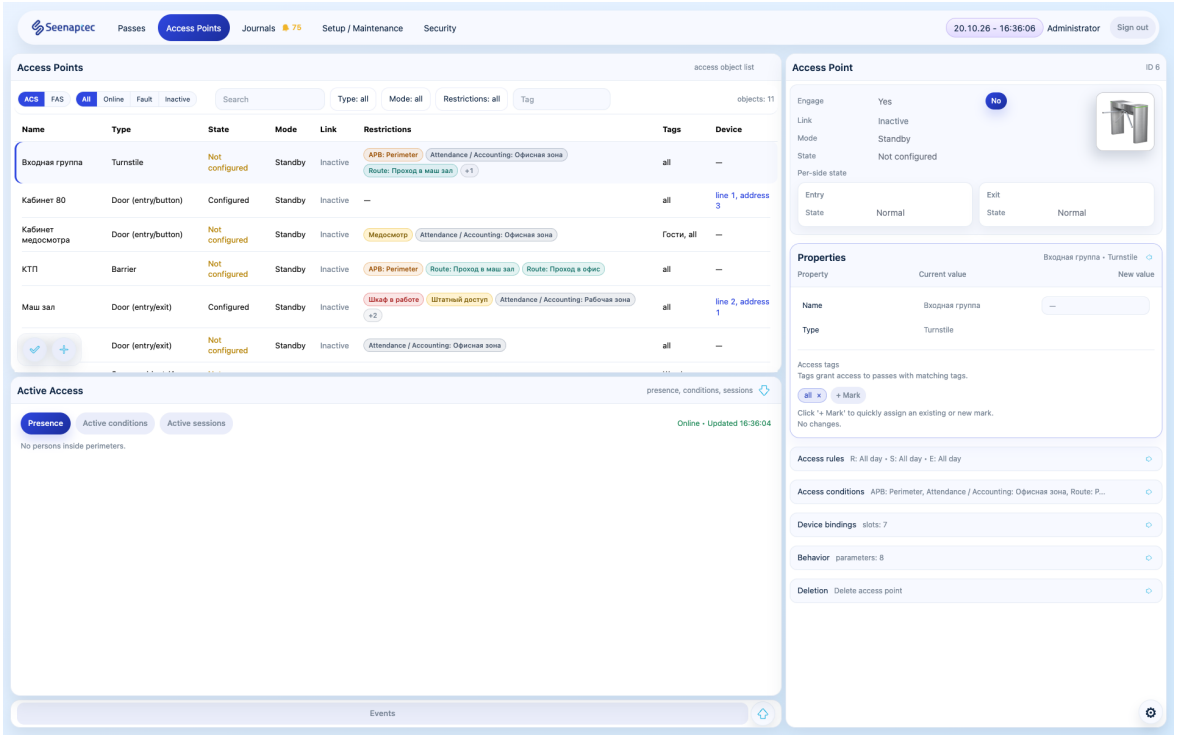
- create an access object before hardware installation;
- replace a field device without reassigning access permissions;
- preserve the object's settings and event history when replacing hardware.

The object type is selected at creation time and determines the set of functional slots, available parameters, operating modes, and rules.

Object Type	Purpose and Characteristics
"Door (entry/button)"	Reader on entry and exit button inside. Slots: reader, lock, exit button, door contact sensor, emergency release sensor. Supports APB rules and door control modes.
"Door (entry/exit)"	Credential-based passage in both directions; no exit button. Slots: entry and exit readers, shared lock, door contact sensor, emergency release sensor. Direction and feedback are determined by the reader where the credential was presented. Supports APB rules and door control modes.
"Turnstile"	Separate slots for entry and exit readers and actuators, plus a passage sensor. Records the passage outcome: "Passage completed" or "No passage occurred".
"Server cabinet (1 door)"	Slots: reader, lock, door sensor, handle sensor. Operation is tracked as an access session—with start, end, and latest confirmation timestamps.
"Server cabinet (2 doors)"	Front and rear doors sharing a single enclosure. Each side has its own reader, lock, door sensor, and handle sensor. Access permission applies to the cabinet as a whole. Tracked as a single session: opening either side starts the session; closing both sides ends it. The side is recorded in events and state as "Front" or "Rear".
"Barrier"	Single boom arm for both entry and exit. Slots: entry and exit readers, boom drive, passage sensors; optionally a safety sensor and boom position sensor. Drive control mode and sensor configuration are defined by object parameters. Records the passage outcome.

Access permissions are assigned to the access object, not to a controller, reader, or other hardware component. Once components are bound, the object processes access requests and records associated events.

11.2. “Access Objects” Panel



List columns: “Name”, “Type”, “State”, “Mode”, “Link”, “Restrictions”, “Tags”, and “Device”. Each column represents a distinct aspect:

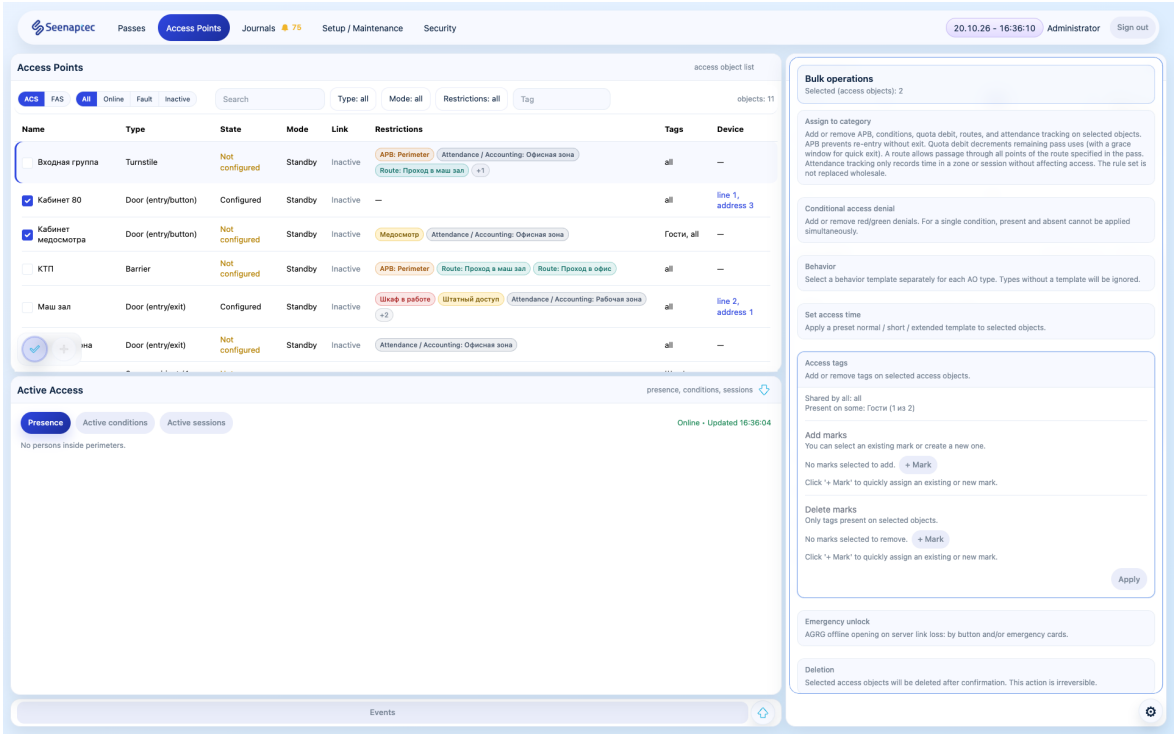
- **“State”** — configuration readiness: “Configured” (all required slots are bound), “Not configured”, “Inactive” (operation of the access object has been manually disabled);
- **“Mode”** — current operating mode: “Standby” (**Normal / Duty**), “Access open” (**Access granted**), “Handle open” (**Handle unlocked**), “Unlocked”, “Blocked” (**Locked**); alarm modes are highlighted in color;
- **“Link”** — aggregate connectivity across the object’s required devices: “Active”, “Offline”, “Inactive”;
- **“Restrictions”** — badges for assigned rules (perimeters, conditions, quota debit);
- **“Tags”** — the first several tags assigned to the access object;
- **“Device”** — bound device; clicking it opens the device in “Setup / Maintenance”, and you can return to the access objects list using the back button in the panel header (Section 8.10).

Above the list are quick status filters (**All | Online | Fault | Inactive**), search, and filters by type, state, mode, link, restrictions, and tags.

Creation. The “Add access object” button opens a dialog with “Name”, “Unique code”, “Type”, and access tags. Batch creation from a device list is performed in “Setup / Maintenance” via the “Bulk create/bind access objects” button. Each row specifies a code or name, device serial numbers, object type, and binding template—entered manually or imported from CSV. After validation, the objects are created and bound in a single batch.

Selection mode enables bulk operations on access objects: assigning categories, denying access by condition, applying a behavior template, setting an access time template, managing tags, configuring emergency unlock, and deleting objects.

The **“Emergency unlock”** bulk operation enables or disables offline emergency unlock across the selected objects: choose “Enable” / “Disable” and independently check “By button (offline)” and “By emergency cards (offline)”. Objects that do not support the operation (an object type without emergency unlock, an incompatible binding topology, or device firmware earlier than 2.0.0) are filtered out of the batch with an explanation. The operation is applied asynchronously: check the result in the “Emergency unlock” section of the access object card (Section 11.3).



11.3. Access Object Card

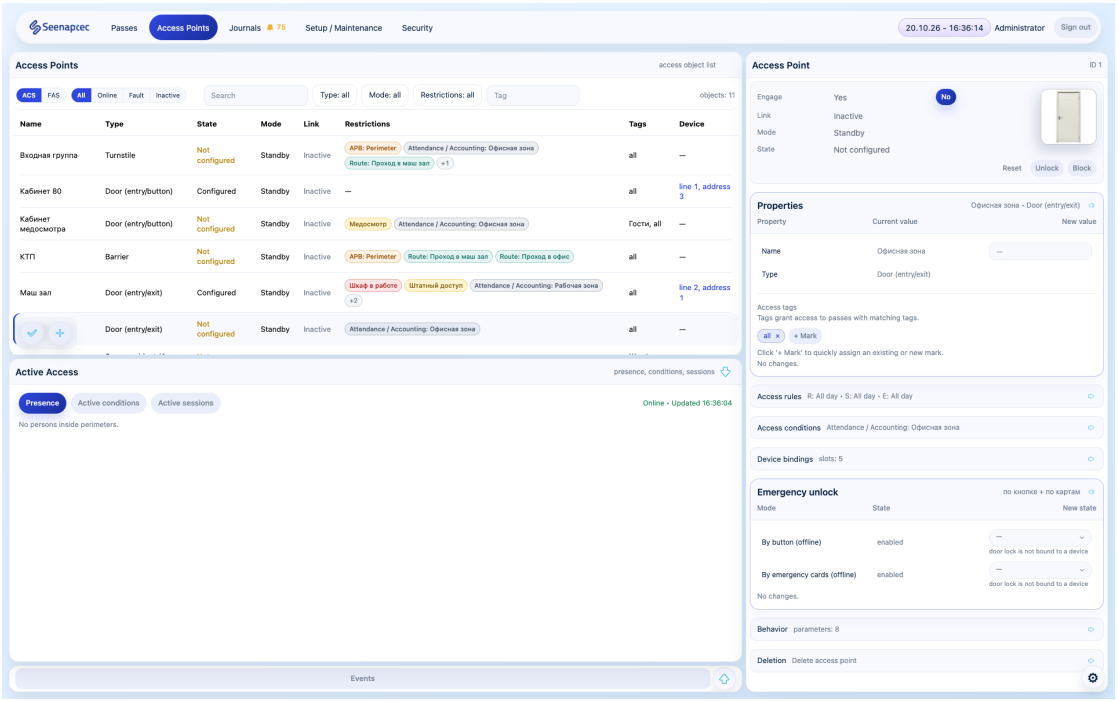
The top block of the card shows operational status: “Engage”, “Link”, “Mode”, and “State”. For object types that support operator modes (doors, cabinets, etc.), direct mode controls are also located here: “Unlock” (continuously open for passage), “Block” (access denied), and “Reset” (return to normal standby/duty mode). Commands execute asynchronously; mode transitions are confirmed by the object state and a journal event. There is no single-pulse open button in the UI: one-time access is granted by presenting an identifier.

Card sections:

- **“Properties”** — name, type, and access tags: “Tags grant access to passes with matching tags”—a mechanism for granting permissions to a group of objects via a single tag (Chapter 3). > **For administrators familiar with traditional ACS platforms.** A tag replaces an “access group” (“access level”), but works in the opposite direction. In a traditional system, an access group is a fixed list of doors written into card settings: if the door list changes, cards or controller levels must be rewritten. Here, the tag lives on the access object: a pass is granted access “by tag”, while group membership is managed on the objects—attach the tag to a new door, and all passes carrying that tag gain access to it automatically. For directional object types, tag-based access can be narrowed by an **access class**: for example, a tag on turnstiles can be restricted to “entry only” or “exit only” (the class is selected when issuing the pass, see Section 12.4). A priority rule applies: **if an access object is listed explicitly in a pass, only that explicit entry governs that object**—tags are ignored for it; a tag grants access only to objects not explicitly listed in the pass.
- **“Access rules”** — identification mode and access hours (Section 11.4).
- **“Access conditions”** — categories and condition-based denials from the rules catalog (Section 11.4).
- **“Device bindings”** — mapping of the object’s functional slots to device components (Section 11.5).
- **“Behavior”** — operating parameters: timeouts and warnings; populated manually or via a behavior template, and current settings can be saved as a template for objects of the same type. For each input slot (“Door sensor”, “Exit button”, etc.), an **“Inversion”** flag is available to invert the interpretation of the input signal—allowing you to wire a sensor with opposite polarity without reconfiguring the hardware device. By default, inversion flags are off: buttons operate as normally open (active on close) and door contact sensors operate as normally closed; that way, a broken button wire does not unlock the door, while a broken door sensor wire is detected as a forced-open alarm. Keep the converse in mind: enabling button inversion for a normally closed contact means a wire break will trigger door opening. Slot inversion combines with the point-level input polarity configured on an AGRG device (Section 10.4): enabling inversion at both

levels cancels out. Additional parameters: **“Door sensor locking delay”** (or **“Passage sensor locking delay”** for a turnstile) — a pause of `0` to `5000` ms between sensor closure and lock re-engagement (`0` by default); **“Authorized exit by button”** (single-sided doors only) — personalizes exit by button: after the exit button is pressed, the system waits for a card presentation on the entry reader before the door closes and logs the exit under that specific subject (disabled by default); **“Authorized exit sound signal”** — confirms the card presentation with a reader beep (enabled by default).

- **“Emergency unlock”** — configures offline emergency unlock for the object on AGRG devices. This section is available on firmware version 2.0.0 and above (earlier versions show a placeholder prompting a firmware update) and requires a compatible binding topology: the object’s reader and lock must be bound to the same device, and that device may host at most one lock-equipped access object. Two independent aspects are configured in the rows “By button (offline)” and “By emergency cards (offline)”: the “State” column shows the current database value, “New state” selects “enable” / “disable”, and changes are saved with “Save” / “Reset”. Enabling “By emergency cards (offline)” requires emergency identifiers assigned to the object—up to three cards of the “Emergency” type (Section 12.6); disabling is always available. Hardware application is confirmed via echo: the row status shows “applied”, “pending apply”, or “apply error” (check the journal for the error reason). Emergency unlock activations that occur on the device while it is offline are delivered to the journal once connectivity with the server is restored (Section 13.1).



- **“Deletion”** — deletes the access object and clears its hardware bindings.

The card may also include a “Monitoring” (video) section. It and other video-related card elements—camera binding and video recording parameters in the “Device bindings” and “Behavior” sections—are displayed only when video integration is configured, i.e., when a video server address is set in the “Video server” section of the “Setup / Maintenance” workspace (Chapter 10). Video integration itself is not covered in the current edition of this guide.

11.4. Object Access Rules

An object’s access rules are organized into two adjacent card sections: “Access rules” covers identification and schedules, while “Access conditions” covers categories and condition-based denials. Each section has its own “Reset” and “Save” buttons.

Identification and restrictions:

- “Identification mode” — “Card only”, “Card + PIN”, “Card + escort confirmation”, or “Card + PIN + escort confirmation”. PIN modes require a “Confirmation PIN code” to be set on the person (Chapter 12). Escort confirmation modes implement two-person control for persons marked with “Requires escort confirmation” (Section 12.3): after such a person presents their card, the system waits 15 seconds for an escort’s card—an employee marked with “Can confirm escort” who also holds valid access to this object. Other persons pass through the point normally: using a card in “Card +

escort confirmation" mode, or a card and PIN in "Card + PIN + escort confirmation" mode. Escort confirmation applies identically on entry and exit. At an escort-controlled point, only the confirming escort enters a PIN—following the direction rules: on exit, a personal PIN is not requested, so the confirming escort presents only their card;
- "Access under duress" — behavior when a duress PIN is entered (a code that differs from the person's normal PIN by incrementing the last digit by one): "Allowed" — passage is granted indistinguishably from normal access; "Denied" — access is refused. In both cases, the event is flagged as a duress passage and raises a critical-severity alarm.

Time profiles (for object types that support schedules): a weekly access grid across three profiles—"Regular hours", "Shortened hours", and "Extended hours". Which profile applies to a given subject is determined by their pass ("Access time" field, Chapter 12); passes set to "Unlimited" are allowed 24/7. The grid is populated manually or from an access time template; the "Use production calendar" toggle restricts access on weekends and public holidays (no access) according to the calendar directory (Section 11.7). Current profiles can be saved as a template.

Categories and denials ("Access conditions" section) — assigns rules from the shared catalog (Section 11.6) to the object:

- "Belongs to category" — includes the object in an APB perimeter, sets or clears a condition upon passage, or debits pass quota ("Quota: …");
- "Deny access when present" — denies access to subjects who currently have the specified condition (a blocking condition);
- "Deny access when absent" — denies access to subjects who lack the required condition (such as safety briefing or special clearance).

11.5. Device Bindings

Each object type defines a set of **slots**—functional attachment points for device components: for example, a door has "Entry reader", "Lock", and "Exit button" (required), plus "Door sensor" and "Emergency release button" (optional); a turnstile has entry and exit readers and locks plus a "Passage sensor"; a cabinet has "Reader", "Lock", "Door sensor", and "Handle sensor".

The recommended binding method is via a **binding template**: a preconfigured "device component → object slot" mapping for a standard wiring scheme (templates are managed in the system tools of "Setup / Maintenance", see Chapter 10). In the "Device bindings" section, the "Binding mode" selector offers templates compatible with the object's type: when a template is selected, slot-by-slot binding is replaced by assigning devices to template groups (with an "Assigned: N of M" summary). Templates are also used when creating objects from selected devices, including batch creation (Section 11.2): a matching template is selected automatically, and manual slot binding is hidden.

"Manual" mode is an alternative for custom wiring schemes: you select a specific device component from a line for each individual slot. A manually assembled scheme can be saved as a template ("Save current binding as template"); complex specialized bindings cannot be edited through a template.

Where binding is performed. Mapping "device component → object slot" is the same underlying operation accessible from several places in the UI; choose whichever starting point best fits your workflow:

- **access object card**, "Device bindings" section (this section) — when starting from the logical object: configuring a new door or changing its wiring scheme;
- **device card** in "Setup / Maintenance", "Access object binding" section (Chapter 10) — when starting from the hardware: the engineer views the device's components and connects them to object slots;
- **"Create/bind access object" button** in the header of the "Equipment" panel (the access object icon, active when a device is selected) — opens a dialog with "Create" (new object bound via template or manually) and "Bind" (existing object) modes. If the device is already bound, the dialog offers **"Unbind"** with confirmation in the right panel; unbinding preserves the access object itself and only breaks its link to the device. While a device is bound, creating or binding another object from it is unavailable;
- **bulk operations in the "Equipment" panel**, "Create/bind access object" section (Section 10.5) — creates and binds objects for multiple devices selected in the list;
- **batch creation** — the "Bulk create/bind access objects" button in the "Equipment" panel (Section 11.2): batch creation and binding from rows of device serial numbers or a CSV file.

All of these workflows use the same binding templates from the "Setup / Maintenance" system tools (Chapter 10), and the result is reflected identically in the object's state and recorded in the audit log.

In the access object card and device card, bindings can be saved incrementally even when the set of required slots is incomplete: partial progress is saved, including unbinding previously bound slots. Bulk operations are stricter—batch creation and binding

require a complete set of required slots immediately.

The same distinction applies when changing a behavior parameter that alters which slots are required—for example, changing the passage sensor mode on a barrier or turnstile. In the access object card, such a mode change is saved immediately even if the newly required components are not yet bound: the behavior section displays a yellow warning listing the missing bindings, which disappears once the required components are bound. In bulk operations (batch application of behavior or time templates), the same mode change is rejected with an error if required bindings are missing.

As long as any required slots remain unbound, the access object stays in the "Not configured" state and does not control hardware.

11.6. Rules Catalog: Conditions, APB Perimeters, and Quota Debit

The **"Access rules"** section of the tool panel is a unified catalog of rules that can be assigned to access objects.

> **For administrators familiar with traditional ACS platforms.** In traditional systems, anti-passback and similar restrictions are controller-level flags tied to a specific circuit board. Here, a rule is a named record in a shared catalog that is assigned to objects like a tag: a single perimeter can span doors across different lines and controllers, a single condition can be checked at any number of access points, and a single quota rule can operate across all building entrances. You create the rule once and add objects to it; all logic executes on the server, so the set of participating objects can be changed without reconfiguring field hardware.

A **Condition** is a named attribute of a subject (for example, "Safety briefing completed" or "Server room clearance"). A condition has a validity period in minutes (or no expiration), a description, and **directional triggers** that define how the condition behaves on objects assigned to its category:

- "Set on" — "passage/access", "entry/access start", or "exit/access end": when a subject performs the specified movement through a category object, the condition is set;
- "Clear on" — "do not clear" or one of the same three triggers: the reverse trigger that automatically clears the condition.

Trigger labels are dual-purpose: the first part ("passage", "entry", "exit") applies to passage objects, while the second part applies to session-based objects (cabinets): instead of a passage, a cabinet has an access session, so "entry" means access start, "exit" means access end, and "passage" triggers on both session boundaries. On passage objects, "entry" and "exit" trigger only on readers with a configured direction, whereas "passage" triggers on any confirmed passage. If both setting and clearing triggers match on the same event, setting takes precedence. The expiration timer continues to act as a safeguard: the condition expires automatically even if an exit was never recorded. Pairing set and clear triggers enables presence-dependent rules: for example, a "Present in workshop" condition is set upon entry through the workshop turnstile, cleared upon exit, and required by doors leading into hazardous zones inside the workshop.

A condition can be:

- set and cleared automatically upon passage through a category object (via the triggers above);
- required ("Deny access when absent") or prohibited ("Deny access when present") on other access objects;
- set and cleared manually in a person's card (Chapter 12).

An **APB perimeter** (anti-passback perimeter) is a zone that prevents repeated passage in the same direction: participating objects define the zone's entries and exits, and the system tracks which subjects are currently inside. Perimeter parameters:

- "Breach mode": "Hard (deny)" — repeated entry or exit without a prior entry is denied; "Soft (log only)" — passage is allowed, but the violation is logged;
- "Re-entry delay, sec" — allows a subject to request passage again within a short window (for example, if they unlocked a door, did not walk through, and let it close);
- "Deny exit by internal card" — allows exit from the perimeter only with regular cards (prevents passing internal cards back to others);
- "Auto-reset, min" — automatically clears presence records after the specified number of minutes.

Quota debit is a rule that turns a confirmed passage through an object into a deduction of one trip from the pass's usage quota ("N of M" limit, see Chapter 12). A quota rule defines a name, a "Grace window, min", and a description; on an access object, it is displayed as a "Quota: ‹name›" badge in the "Belongs to category" block. Quota rules apply only to object types with confirmed passage (both door types, turnstile, barrier).

The grace window defines when exactly a trip is debited after entry. Complete logic:

- **entry** through an object with a quota rule starts a visit: the system waits for an exit within the grace window, and no debit occurs at the moment of entry;
- **exit within the window** debits one trip—the visit is charged upon exit;
- **window expires before exit** — one trip is debited automatically when the grace window expires, even if the subject is still inside or left without being recorded;
- **exit without a recorded entry** debits nothing;
- with a grace window of `0`, every entry is debited almost immediately;
- a single entry debits at most one trip: duplicate signals or a race between simultaneous exit and window expiration never cause a double debit.

Quota is checked on every **entry** access decision: a pass whose quota is exhausted is denied entry. **Exit is always allowed**, even when the quota is exhausted, and does not check the remaining balance—otherwise a subject whose last trip was debited upon window expiration would be unable to leave. Each debit is recorded in the journal as a separate event indicating the trigger ("On exit" / "After grace window expires") and the remaining trip count.

Passes without a finite counter (unlimited passes) are unaffected by quota rules: entry with such passes is not restricted and generates no debits. The entry check uses the available balance—the stored balance minus any pending visits that have started or debited but not yet been committed to the counter—so both quota debits and quota top-ups (Chapter 12) take effect on access decisions immediately.

A **Route** is a named set of passage access objects through which a pass is granted access as if it carried a tag for all points on the route, but without "entry/exit" access classes and without binding to reader direction. A route is assigned to a pass (Chapter 12); on an access object, it is displayed as a "Route: ‹name›" badge in the "Belongs to category" block. For the routes catalog, see Section 11.10.

Presence tracking (time and attendance observation) is a non-intrusive working time tracking mechanism: it records "time in zone" or "session duration" intervals for persons explicitly enrolled in a tracking zone. It **does not affect** access decisions. A "Pres: ‹name›" badge is placed on passage objects and server cabinets; a person's membership in a zone is configured in the person's card (Chapter 12). Closing an interval writes an event to the journal ("Presence tracking" domain, see Chapter 13). For the catalog and monitor, see Section 11.11.

11.7. Production Calendar

The "Production calendar" section of the tool panel is a year-by-year directory of day types: "Workday / Working", "Holiday / Day off", "Holiday", and "Shortened". When internet access is available, a year's calendar can be loaded from public external sources (the `isDayOff` and `xmlCalendar` providers publish the official Russian production calendar) and then adjusted manually. When the network is unavailable, the "Generate" provider populates the year by marking Saturdays and Sundays as days off with no public holidays—the administrator then marks public holidays and transferred working days manually. Access objects with "Use production calendar" enabled apply this calendar when evaluating access rules (Section 11.4): on weekends and holidays, access under regular/shortened/extended profiles is denied.

11.8. "Active Access" Panel

Provides a live operational picture across three tabs (updates automatically; connection status is shown in the header):

- **"Presence"** — who is currently inside each APB perimeter: person, entry time, and entry object. The "Reset presence" (per person) and "Reset all" (per perimeter) buttons perform an operator APB reset—for example, after an evacuation or tracking discrepancy;
- **"Active conditions"** — which conditions are currently active for which subjects and until what time; any active condition can be manually cleared here;
- **"Active sessions"** — currently open access object sessions: object, session type ("Cabinet service", "Opened by operator", "Blocked"), person, and duration. The "Reset session" button forcibly terminates a session—for example, if an engineer walked away without properly closing a server cabinet.

All manual resets are recorded in the journal with the initiating operator.

11.9. "Events" Panel

The panel displays access events and object audit records: "Allowed" / "Denied" decisions with their exact reason, session episodes, violations, and object setting modifications with "before/after" field diffs. The decision reason is shown both in the

event row and in the event card, for example: “Allowed by pass”, “Allowed with escort confirmation”, “Access denied: no permission for this object”, “Invalid PIN code”, “Anti-passback violation: repeated entry”, “Access denied: outside permitted schedule”, or “Access denied: non-working day according to production calendar”. Tailgating or forced entry through an unpermitted open door (Section 4.3) is recorded as an “Unauthorized passage” alarm.

The “Selected” toggle narrows the feed to the currently selected access object—combining its access events and configuration history in a single stream. General event feed controls are described in Chapter 8.

11.10. “Routes” Catalog

The **“Routes”** section of the tool panel is the catalog of named routes. A route is assigned to:

- a pass — in the pass issuance dialog and in the pass card (the pass’s route list);
- an access object — via the “Route: ‹name›” badge in the “Belongs to category” block (Section 11.6); supported only on passage object types (both door types, turnstile, barrier).

Each catalog entry has a name and description. A route can be assigned or removed across multiple selected objects using the “Set route” bulk operation. Deleting a route from the catalog does not strip existing badges from objects or passes—the badges remain visible, though their link to the catalog entry is lost.

11.11. “Presence tracking” Catalog

The **“Presence tracking”** section of the tool panel is the catalog of working-time tracking badges.

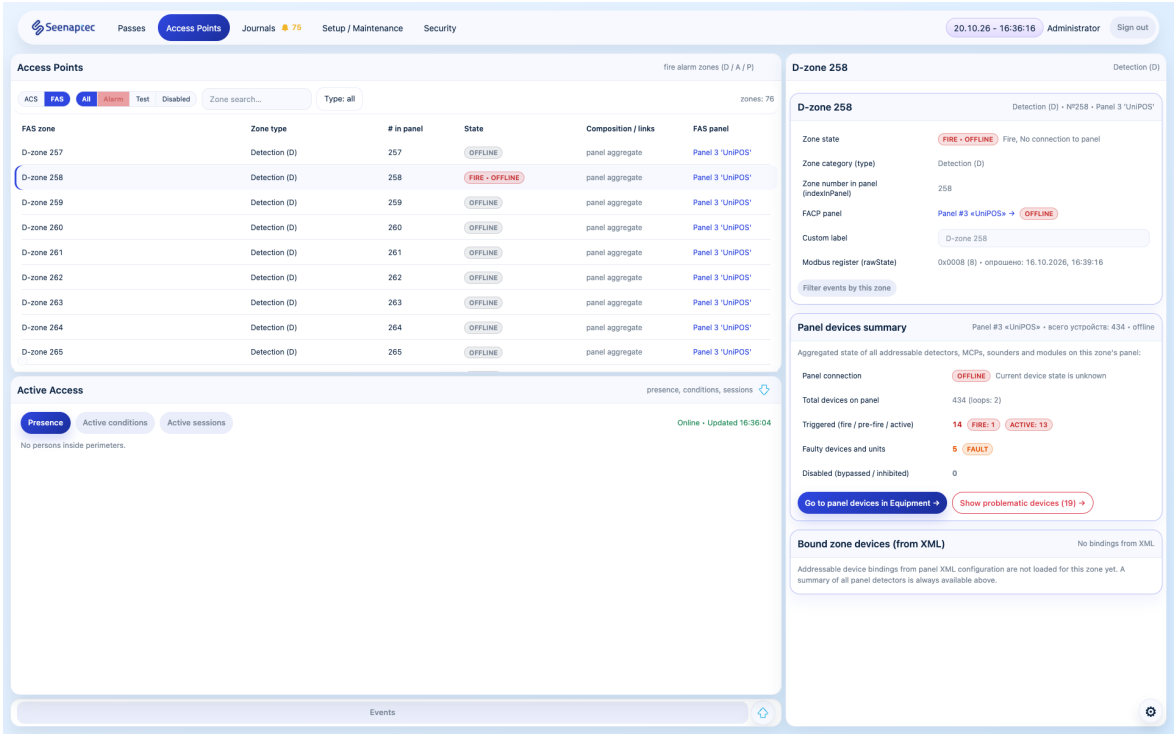
Catalog. Each tracking zone defines a name, description, and “Close timeout, sec”—after this duration, a stale open interval is closed automatically and marked as unreliable in the journal (`0` = never close by timeout). A zone is assigned to:

- an access object — via the “Pres: ‹name›” badge (supported on passage objects and server cabinets);
- a person — in the “Presence tracking” section of the person’s card (Chapter 12): intervals are recorded **only** for persons explicitly enrolled in the zone.

Presence tracking does not replace an APB perimeter (Section 11.8): APB can deny access, whereas presence tracking only observes time intervals.

11.12. FAS Objects (Fire Zones D, A, P)

When the **`ACS | FAS`** pill in the top bar is switched to **`FAS`**, the “Access Objects” workspace (labeled **“Objects”** in the top navigation bar) switches to monitoring and operational control of FAS fire zones. Zones are discovered automatically during a composition scan of connected UniPOS IFS7000 panels and enriched with custom names and detector bindings when a project `.xml` configuration file is imported (Section 10.9).



Left Zone List Panel and Filtering

The left panel displays a unified list of all fire zones across connected panels with the columns `Zone`, `Type`, `Panel`, and `State`. Above the list are quick filtering controls:

- **quick state filter `All | Alarm | Test | Disabled`** — a segmented selector for instant filtering by operational state. Whenever at least one zone in the system is in a fire alarm, pre-alarm, or active output state, the **"Alarm"** button pulses red and displays a counter of alarming zones;
- **search and zone type multi-select** — a text search box matching zone names/numbers and a dropdown filter **`Type: Detection (D) | Notification (A) | Automation (P)`** allowing you to display one or multiple zone types.

If a UniPOS panel loses Modbus TCP communication with the server, all of its zones in the list explicitly display **"OFFLINE"**.

Zone Types and UniPOS State Phases

The platform supports all three logical zone types of UniPOS IFS7000 series panels and distinguishes their hardware state phases:

1. **Detection zone (`D`, Detection)** — groups automatic fire detectors (smoke, thermal, combined, gas) and manual call points (`MCP`) within a protected area:
   - **`Normal`** — normal standby monitoring mode;
   - **`Pre-alarm`** — preliminary activation signal (verification stage or single detector activation prior to full fire alarm);
   - **`Fire`** — confirmed fire alarm in the zone, explicitly indicating the alarm source: **from a manual call point** (`MCP`) or **from an automatic detector**;
   - **`Fault`** — hardware fault, line break, or communication loss with a detector in the zone;
   - **`Test`** — the zone has been placed into detector service test mode by an operator;
   - **`Disabled / Bypass`** — the zone is temporarily excluded from generating alarms.
2. **Notification zone (`A`, Alarm / Notification)** — controls audible, voice, and visual alarm notification circuits and distinguishes three UniPOS panel phases based on the status register bitmask:
   - **`Warning (Fire 1)`** (`warning`, bit `0x0040`) — first alarm stage (local staff alert prior to general evacuation release);
   - **`Start delay`** (`delayed`, bit `0x0010`) — countdown of the programmed delay before activating Stage 2 sounders/strobes;
   - **`Triggered / Notification active (Fire 2)`** (`active`, bit `0x0008`) — active release of sirens and evacuation signs in the zone. `Normal`, `Fault`, and `Disabled / Bypass` states are also tracked for `A` zones.

3. **Automation zone (`P`, Automation / Protection)** — controls fire protection and building automation circuits (smoke extraction, fire dampers, HVAC shutdown, suppression release, and `FPE` modules):
   - **`Start delay`** (`delayed`, bit `0x0010`) — pre-release countdown timer running before activating automation actuators;
   - **`Automation zone triggered`** (`active`, bit `0x0008`) — active release of fire automation equipment in the zone;
   - **`Normal`**, **`Fault`**, and **`Disabled / Bypass`** — normal standby state, release circuit fault, or zone bypass.

FAS Zone Card and Operator Commands

Selecting a zone opens its card in the right area:

- **Status summary and properties** — zone number and editable display name, zone type (`Detection zone (D)`, `Notification zone (A)`, or `Automation zone (P)`), parent FAS panel, current state phase, alarm source, and detailed panel flags.
- **Operational control commands** (available to operators with FAS control permissions):
  - for a **detection zone (`D`)**: **"Arm (Put on guard)"** (returns the zone from test or bypass to normal standby), **"Test mode"** (places the zone into detector walk-test mode), and **"Disable (bypass)"** (temporarily bypasses the zone);
  - for a **notification zone (`A`)** and **automation zone (`P`)**: **"Arm (Put on guard)"**, **"Start (Activate)"** (remotely triggers notification or automation release), **"Stop / Reset"** (stops active release and resets the zone activation), and **"Disable (bypass)"**.
- **Bound devices ("Zone composition")** — a table of all addressable detectors, modules, and `BIO` outputs assigned to the selected zone (populated via XML configuration import or object bindings), showing their addresses, device types, and live states, with one-click navigation to the device card in the "Setup / Maintenance" workspace.

Chapter 12. Passes

The **Passes** workspace is the pass office workstation for managing persons, identifiers, and pass issuance. The left panels are **Passes**, **Events**, and **Identifiers**; the tool panel contains **Access templates**, **Identifier templates**, **Custom personnel fields**, and **Auto-cleanup**.

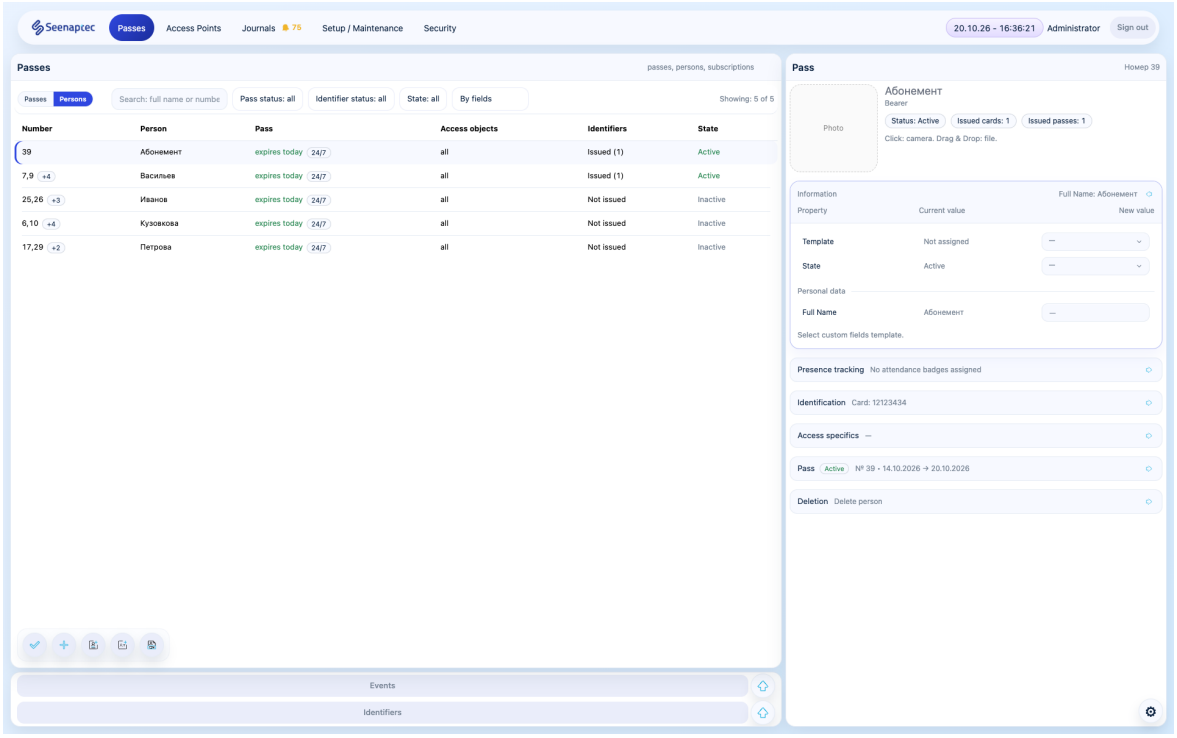
12.1. Core Principle: A Pass Is a Document

> **For those familiar with traditional ACS.** In a traditional access control system, access is defined by an "access level" assigned to a card—the card itself is the permission. Here, the permission is a **pass**: a standalone document that answers every question about access at once—*who* (the person / pass holder), *with what* (bound identifiers), *where* (access objects and tags), *when* (validity period and access time), and *how many times* (quota / uses). A card is only one of the identifiers bound to a pass; a person can hold multiple passes with different permissions and validity periods simultaneously. The role of standard "access levels" is played by **access templates** (Section 12.5)—reusable pass presets—while pre-approval is handled through a **request** (Section 12.4), a draft document that reserves a pass number prior to actual issuance.

A pass in the system is designed as a strict-accountability document:

- its **number** is assigned by the server and is never reused;
- once issued, only the following can be modified: extending the validity period (the end date can only be moved later), topping up the usage quota, binding and unbinding identifiers, changing the escort host, and toggling the state (suspending and resuming);
- the person, start date, access time, and the set of access objects and tags of an issued pass **cannot be changed**—if different access rights are needed, a new pass is issued;
- every action performed on a pass is recorded in the audit journal.

12.2. Passes Panel



The **Passes** panel is used to view persons and passes, search records, create persons, issue passes, and run bulk operations.

Display Modes

The **By persons** (**Passes** / **Persons**) mode switcher changes the list contents:

- when **Passes** is active, each row represents an individual pass;
- when **Persons** is active, each row represents a person, including persons without passes.

When switching between **Passes** and **Persons** modes, the selected person is preserved; in **Passes** mode, the person's current pass is automatically highlighted.

The list displays the following columns:

- **Number**;
- **Person**;
- **Pass**;
- **Access objects**;
- **Identifiers**;
- **State**.

The **Pass** column shows a brief summary of the pass validity period and state, for example:

- "valid until … · N d left";
- "ended N days ago";
- "suspended";
- "unlimited";
- "request";
- "not issued".

The **24/7** badge indicates a pass with unrestricted access hours. Hover over the cell to view the exact validity dates and access time profile.

The **State** column displays the person's current state: **Active**, **Inactive**, **Blocked**, or **Blacklisted**.

Displaying Multiple Passes

In **Persons** mode, the **Number** column lists the numbers of the person's first two passes. If the person has more passes, a **+N** badge appears next to them. Hover over the badge to see the remaining pass numbers.

The validity period and access objects shown in a person's row are taken from a single primary pass selected according to the following priority rules:

1. If there are active passes, the pass with the earliest expiration date is shown.
2. If there are no active passes, but there are pending passes whose validity period has not started yet, the pass with the earliest start date is shown.
3. If all passes have expired, the pass with the latest expiration date is shown.

At equal priority, an active pass takes precedence over a suspended one. An unlimited (non-expiring) pass is ordered after passes that have a specific expiration date.

A pass request is shown as a person's primary document only if no passes have been issued. The complete list of a person's passes is always available in the person card under the **Pass** and **Additional pass** sections.

Search and Filtering

The following tools are available for filtering records:

- text search by pass number, full name, or title;
- validity filter: **Current**, **Expired**, **Future**;
- upcoming date filter: **Today**, **Tomorrow**, **This week**;
- filter by access object;
- filter by access time;
- filter by pass status: **Active**, **Suspended**, **Pending**, **Expired**, **Request** (draft).

To apply a value-list filter, open the dropdown, select the desired option, and click the confirmation button.

In **Persons** mode, the following filters are additionally available:

- issuance status filter: **Issued (N)**, **Revoked**, **Never issued**;
- the **By fields** button for searching across custom personnel fields.

Custom field search supports up to three conditions. A configured set of conditions can be saved as a reusable search template.

Creating Persons and Passes

At the bottom of the panel, the multi-select mode toggle button is followed by record creation buttons:

1. **New pass** — creates a pass or a pass request. If a person or their pass is currently selected in the list, the person's details are pre-filled automatically.
2. **New person** — opens the person creation dialog (with the option to proceed directly to pass issuance via the **Create and issue pass** button).
3. **Issue pass to selected person** — opens the pass issuance dialog for the person currently selected in the list. This button is disabled if no person is selected or if the selected person is **Blacklisted**.

A person's complete list of passes is always visible in their card (**Pass** and **Additional pass** sections, see Section 12.4).

The availability of buttons, columns, and sections depends on the system operator's RBAC permissions. If an action is not permitted, the button is disabled with a tooltip explaining the reason; if an operator has read access to persons without read access to passes, the interface operates as a personnel directory without pass and identifier columns or sections. For operators with template-only issuance permissions, the pass issuance workflow is restricted to allowed access templates (Section 12.5).

Bulk Operations and Deletion

To select multiple records, enable multi-select mode. Available actions will appear in the bulk operations panel.

In **Persons** mode, the bulk operations panel contains the **Pass issuance**, **Attendance zone setup**, and **Deletion** sections. In the **Pass issuance** section, select an access template (Section 12.5), verify its contents (validity period, access time, quota, tags,

and access objects), and click **Apply**—new active passes based on the template will be issued to the selected persons, with progress displayed in the **State** column of each row.

Deletion is performed:

- for a single record — from the person card or pass card;
- for multiple records — via the bulk operations panel.

Before deleting, the system prompts for confirmation and lists all affected records.

12.3. Person (Pass Holder)

> **For those familiar with traditional ACS.** A traditional "employee card" has a fixed set of fields identical for everyone: employee ID, department, job title. Here, only the core attributes of a person are fixed (full name / title, photo, state), while the rest of the card is defined by a **personnel field template** (Section 12.8): the organization defines its own custom fields—their names, categories, value types (string, integer, boolean, enumeration with a custom list of options), and whether they are required or unique—and assembles them into templates for different categories of people. An employee gets an employee ID and department; a visitor gets an ID document and host; a contractor gets a company name and contract expiration date. Switching a person's template never erases field values filled outside the new template.

A person (pass holder) is created in the two-column **New person** dialog: on the left, you specify the photo, **Full name / title**, type (**Person** for an individual, or **Bearer** for an anonymous pass holder), and state; on the right, you select a personnel field **Template** (Section 12.8) and fill in the custom fields of the selected template (required fields are marked with an asterisk). The **Create** button saves the person card and template field values; the **Create and issue pass** button saves the person and immediately opens the pass issuance dialog for them (Section 12.4). If the **Blacklisted** state is selected or the operator lacks pass issuance permissions, the **Create and issue pass** button is disabled.

Photo. The photo field (in both the creation dialog and the person card) can be populated in two ways: clicking it opens the computer's camera capture, or you can drag and drop an image file onto the field. The image is automatically resized to 640×480 pixels; if direct camera access is unavailable, the dialog prompts you to select a file from the device. Hovering the cursor over a photo—in the person card, in the identifier card, or when selecting a person in the issuance dialog—smoothly zooms it (up to 3× or up to the window boundaries); the enlarged preview disappears as soon as the cursor leaves the thumbnail.

The person card (shared with the pass view) consists of the following sections:

- **Information** — state (**Active**, **Suspended**, **Blacklisted**), personal details, and **custom fields** defined by the assigned field template: organizational data, contacts, technical metadata, and other categories. The set of custom fields is configured in the **Custom personnel fields** section of the tool panel (Section 12.8).
- **Identification** — the person's cards: adding, revoking, and reissuing identifiers (Section 12.6).
- **Access specifics** — parameters that affect access decisions (**Rules**):
  - dual-control confirmation at high-security access points: **Requires escort confirmation** — the person can pass through a point operating in **Card + escort confirmation** mode (Chapter 11) only after a confirming person presents their card; **Can confirm escort** — the person is authorized to confirm passage for others, including acting as an escort host for visitors (Section 12.4). These flags are mutually exclusive: a person either requires confirmation themselves or confirms others—saving a card with both flags enabled is rejected with an error;
  - **Confirmation PIN code** — a 4-to-12-digit PIN for access objects operating in **Card + PIN** mode; only the "PIN set" status indicator is displayed, and the PIN itself cannot be read back;
  - person conditions — named attributes (clearances, safety briefings) checked by access object rules (Chapter 11);
  - **Presence tracking** — working time zones in which the person participates in presence interval tracking (Section 11.11): intervals are recorded only when membership is explicitly assigned; this does not affect access rights.
- **Pass** — the person's currently selected pass (Section 12.4); any other passes held by the person are shown as collapsed **Additional pass** sections. Each pass section header displays a color-coded status indicator (**Active**, **Suspended**, **Pending**, **Expired**, **Request**), and when collapsed, a summary in the format `#N • period`.
- **Deletion** — a collapsed section at the bottom of the card: deleting a person revokes their passes and unbinds their identifiers, while preserving historical events in the journal. Clicking the delete button requires confirmation in the right panel. Bulk deletion of persons is performed in multi-select mode via the bulk operations panel.

The **Blacklisted** state blocks the person: passes cannot be issued to them.

12.4. Issuing a Pass and a Pass Request

The **New pass** and **Issue pass to selected person** buttons open the unified issuance dialog. At the top is a document type switcher: **Pass** | **Request** (draft).

A **Pass** is issued immediately: on the left are the person (existing or created on the spot) and card; on the right is the **Access** section: start and end dates, **Access time** (**Unlimited**, **Normal**, **Reduced**, **Extended** — duration profiles, see Chapter 11), **Quota / Uses** (passage limit), **Escort host (visitor access)**, tags, routes, and access objects (for two-sided objects, access can be restricted to a specific direction or side: **Entry** / **Exit**, or **Front** / **Rear** for a two-door server cabinet). Dates are kept consistent automatically: if you set the start date later than the end date, the end date is pulled forward to match the new start date (an unlimited end date is unaffected).

When the dialog is opened via **Issue pass to selected person** (or after clicking **Create and issue pass** in the person creation dialog), the person in the left column is locked as read-only: the photo, **Full name / title**, and type cannot be edited, preventing accidental overwrites of the person's record during pass issuance. Below the **Identification** heading, the name of the person's assigned field template (or person type) is shown, and an expandable **Summary** displays the template and the person's populated custom fields.

Escort host (visitor access). The **Escort host (visitor access)** field assigns an employee responsible for the visitor to the pass. At an access point where the visitor does not have independent access rights, the flow works as follows: the visitor presents their card first, opening a 15-second confirmation window; the escort host then presents their own card at the same reader, and access is granted. Confirmation succeeds only if, at the moment of presentation, the escort host has the **Can confirm escort** attribute enabled and holds valid access to that access object. Escort sponsorship only covers a missing access grant: denials for other reasons (expired pass, anti-passback violation, unmet conditions, exhausted quota) are not overridden by the escort host. At access points where the visitor has their own grant, they can pass independently without an escort.

Tags, routes, and access classes. A tag grants the pass access to all access objects carrying the matching tag (Chapter 11). A **route** is a named set of transit access points: the pass receives access to all points along the route without entry/exit class restrictions (Section 11.10); in the issuance dialog, routes are selected in the **Access** section alongside tags and access objects. If the tagged objects include directional types, a **Tag access class (default — all directions)** block appears below the selected tags: for the **Turnstiles** group, the **Entry** / **Exit** buttons narrow tag-based access to a single direction—for example, for a single-use "exit-only" pass. Keep the priority rule in mind: an access object added to a pass explicitly (in the **Access objects** list) is governed solely by its explicit setting—tag direction restrictions do not apply to it.

Person and card in the dialog. While the person field is empty, reading a card automatically fills in its current holder. If a person is already selected or entered, selecting a card that belongs to another person triggers **reassignment (revocation from previous holder)**: the card field displays the notice **Will be revoked from '…'**, and upon saving, the card is unbound from the previous holder and assigned to the person of the current pass (the person in the dialog remains unchanged). Changing the person or card does not reset the selected access template or the contents of the **Access** section. If the card is explicitly pinned to a specific pass of its previous holder, saving will fail with an error—remove the card from that pass first.

Instead of filling in fields manually, you can select an **Access template**, which populates the validity period, access time, quota, and access set (Section 12.5). The first few templates appear in the top bar of the dialog as quick-select buttons—as many as fit the width—while the rest are grouped under the **more…** dropdown. The template's period rule is converted into concrete dates at the moment of selection, after which all fields can be adjusted manually (except in restricted issuance mode, where the access scope is strictly locked to the template).

A **Request** (draft pass) reserves a pass number and holder details without granting access: only the validity period and a mandatory comment are specified, while access objects and access time are assigned later upon issuance. A request appears in the list with the **Request** status; its card provides the following actions:

- **Issue pass** — opens the **Issue pass for request #N** dialog: you assign access objects and access time, converting the request into an active pass;
- **Reject** — deletes the request and records the action in the journal.

Typical scenario: a reception desk or external system creates a pass request, and the pass office issues the actual pass.

Issued pass card (**Pass** section): the pass state is indicated by a colored badge in the section header; editable fields include the pass number, validity period (only the expiration date can be extended, including making it unlimited), **Quota / Uses** (`N of M` — can only be increased), **By identifiers** binding (restricting the pass to a specific card held by the person; selecting `– None –` removes the restriction so the pass works with any active identifier belonging to the person), and the escort host. Access time, tags, and access objects are displayed as read-only—in accordance with the document principle (Section 12.1).

The **Number** row contains a **Pass #N** link: in the primary **Pass** section, clicking it highlights the pass row in **Passes** mode; in an **Additional pass** section, clicking it selects that pass and promotes its section to the primary **Pass** section. Editing document fields (dates, quota, identifiers, escort host) is available only in the primary **Pass** section; **Additional pass** sections display the pass in read-only mode and provide actions to suspend/resume, delete, or switch to that pass.

Suspending and resuming a pass are performed via the **Suspend** / **Resume** button in the bottom action bar of the section (next to **Delete pass**); both operations require `full` permissions on passes and take effect immediately without saving the card. For a pass whose validity period has not started yet (**Pending**), suspension is unavailable; for a pass request, the **Reject** button is shown instead.

Revoking a pass means deleting it via the red **Delete pass** button at the bottom of the pass section (available in both **Pass** and **Additional pass**); the confirmation prompt names the specific document—"Pass #N will be deleted (revoked)". This requires `full` permissions on passes. The pass history is preserved in the journal, and its number is never reused. For a pass request, **Reject** replaces the delete button. Bulk deletion of passes is performed in multi-select mode via the bulk operations panel.

12.5. Access Templates and Restricted Issuance

The **Access templates** section of the tool panel manages reusable pass presets. A template stores:

- **Period rule** — **For today**, **For tomorrow**, **For a week**, **Until date**: the rule is resolved into concrete dates at the time of issuance;
- **Access time** and **Quota / Uses**;
- access objects and tags;
- the **Restricted issuance** flag — authorizes operators with the **Issue by template** (`issue_tpl`) permission level to use this template.

Restricted issuance is designed for reception desks and visitor registration: an operator with the **Issue by template** permission level (Chapter 14) can issue passes **only** using templates that have this flag enabled. In the issuance dialog, such an operator selects an allowed template, a person, and an existing card; the access scope is displayed as read-only and cannot deviate from the template. Setting or clearing the **Restricted issuance** flag requires `full` permissions on templates—so the security officer controls exactly which templates are available at the reception desk.

12.6. Identifiers Panel

The **Identifiers** panel tracks access cards independently of issued passes. An identifier is used by the system to recognize a person, whereas access rights are determined by the linked pass.

The list displays the following columns:

- **Code**;
- **Valid from**;
- **Valid until**;
- **Full name**;
- **State**.

Possible identifier states:

- **Active** — the identifier is assigned to a person and ready for use;
- **Inactive** — the identifier is created in the directory but not assigned to any person;
- **Suspended** — usage is temporarily blocked;
- **Lost** — the identifier is marked as lost and automatically unbound from the person;
- **Revoked** (Withdrawn) — the identifier is permanently decommissioned;
- **Pending** — the validity period has not started yet;
- **Expired** — the validity period has ended.

The identifier kind is shown as a badge in the list, and you can filter by kind. The **Emergency** kind designates an emergency offline unlock card: such a card unlocks an access object when the controller loses communication with the server, provided emergency card unlocking is enabled on that access object (Section 11.3); at most three emergency cards can be active per access object. The **Emergency** badge is highlighted in red.

Creating an Identifier

An identifier can be created in two ways.

From the Identifiers Panel

1. Click the + button.
2. In the **Add identifier** dialog, specify:

- **Template**;
- **Code**;
- **Kind**: **Regular**, **Internal**, **Control**, or **Emergency**;
- start date and validity period.

3. Save the record.

The newly created identifier receives the **Inactive** state and is not bound to any person. This method is used to pre-populate a stock of cards in the pass office.

To assign the identifier to a person later, open its card and click **Issue to current**, or select it when issuing a pass.

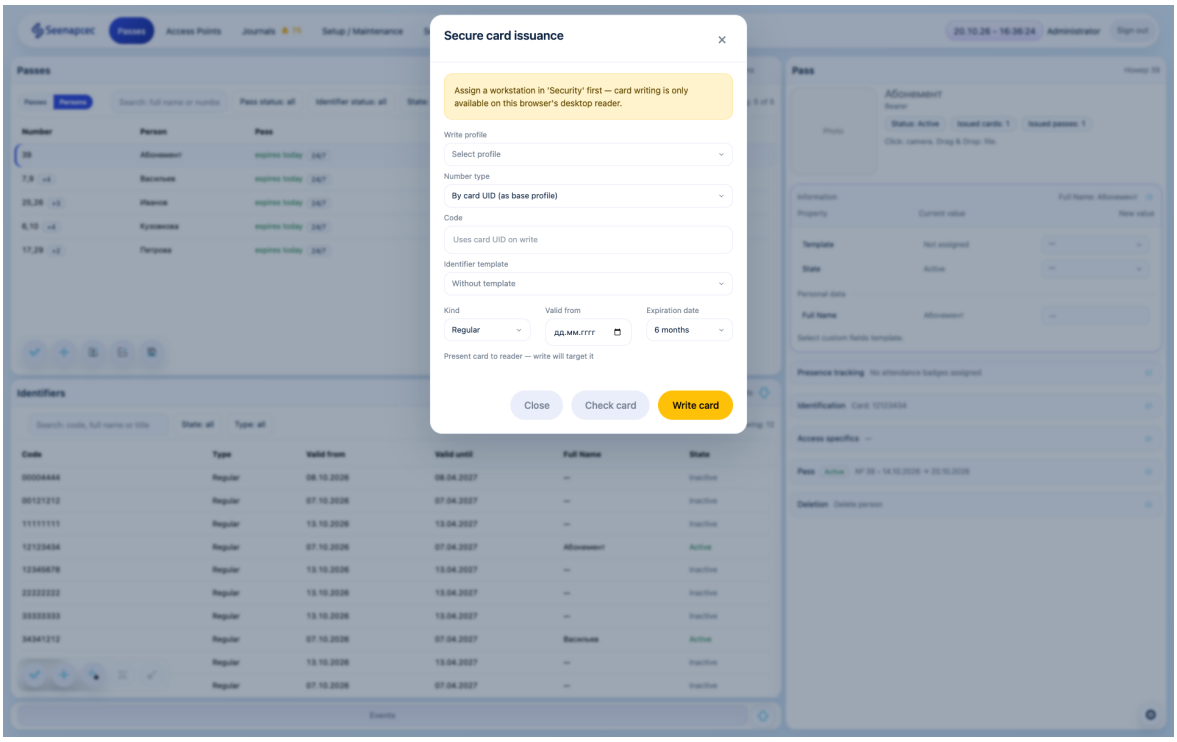
From a Person Card

1. Open the person card.
2. In the **Identification** section, click the + button.
3. In the **New identifier** block, specify the template, code, kind, start date, and validity period.
4. Click **Save**.

The identifier is created and immediately bound to the selected person. Clicking **Reset** closes the block without saving.

Secure Card Issuance (Emission)

Secure card issuance writes an identifier code into the protected memory area of a card on a desktop reader while simultaneously creating the identifier record in the system. It is performed at a Reader Workstation (card agent, see Section 14.8) using a card profile from the library (Section 14.9); `full` permissions on passes are required.



1. In the **Identifiers** panel, click **Secure card issuance**.

2. Select a **Write profile** — a MIFARE Classic, MIFARE Plus, or MIFARE DESFire profile from the library.
3. Select the **Number type** — the source of the identifier code. In all modes, the code is written into the protected area of the card; the modes differ in how the code is generated:
   - **By card UID (as base profile)** — the code matches the card UID (`uid_mirror`);
   - **Manual** — the code is entered by the operator (`stored_code`);
   - **Bind to UID** (**UID-derived**) — the code is derived from the card UID via SHA-256 (`uid_derived`), does not match the UID, and can only be read by readers configured with the secure profile.
4. Specify the identifier parameters—template, kind, start date, and validity period—just as in the **Add identifier** dialog.
5. Place the card on the desktop reader and click **Write card**. Keep the card on the reader until the operation completes—progress is shown in the dialog's message feed.

The **Check card** button probes the card state without writing—for any profile (MIFARE Classic/Plus and DESFire): the dialog displays the card's security level and current state and indicates whether writing with the selected profile is possible. If the card is not ready for writing (a factory MIFARE Plus card, a DESFire card without the profile application, or a corrupted DESFire structure), the dialog offers a direct link to Card Initialization, where card preparation is performed.

Secure card issuance requires: the Reader Workstation agent to be online, the browser to have that Reader Workstation assigned (Section 14.8), and the profile keys to be stored in the secrets vault (Section 14.5)—otherwise, the dialog will report that keys were not found.

Identifier Actions

The following actions are available in the identifier card:

- **Claim / Take** — unbind the identifier from its current person;
- **Issue to current** — bind the identifier to the person currently selected in the **Passes** panel;
- **Block** — suspend usage of the identifier;
- **Unlock** — restore usage of a blocked identifier;
- **Reissue** — replace the identifier with a new one while preserving all existing bindings;
- **Delete identifier** — delete the record after confirming the operation.

The availability of actions depends on the identifier's current state and the operator's permissions.

Identifier Reissue

The **Reissue** operation replaces a lost or damaged card with a new code without modifying the person's passes.

1. Open the identifier card or the person card.
2. In the **Identification** section, click **Reissue**.
3. Enter the new code manually or scan it with a desktop/control reader.
4. Verify the **Will be unbound** badge next to the previous identifier code.
5. Click **Save**.

While identifier reissue is in progress, other fields in the section are locked.

Upon saving:

- the previous identifier transitions to the **Revoked** state and is unbound from the person;
- a new identifier record is created with the new code;
- the identifier kind and its bindings to the person and passes are preserved;
- the corresponding revocation and issuance events are recorded in the journal.

The new identifier receives the same validity duration as the previous one, with its start date set to the reissue date. An unlimited identifier remains unlimited.

Card Reissue

Card reissue writes a duplicate of an existing identifier code onto a new physical secure card—without changing the identifier directory (unlike identifier reissue, which creates a new identifier with a new code). The **Card reissue** button in the **Identifiers** panel becomes active when an identifier card is open and requires `full` permissions on passes.

1. Open the identifier card and click **Card reissue**.

2. Select the **Write profile** — the target card profile for the new card.
3. Place the card on the desktop reader. Clicking **Check card** probes the card without writing: the dialog displays the card state and indicates whether it can be written with the selected profile.
4. Click **Write card**. Keep the card on the reader until the operation completes.

In the standard scenario (when the card is already initialized for the target profile), the card keys are not changed: if the target key does not match, writing is rejected and the card remains untouched.

The **with key rotation (re-key)** scenario is available if the operator holds the **Card initialization** permission (Section 14.2): the **Source card profile** field appears in the dialog. Select the profile corresponding to the card's current keys—for a new MIFARE transport card, this is a transport profile from the library. If the target key does not match, the card agent authenticates with the source key, rotates the sector keys to the target profile keys, and writes the code; the outcome ("Written with rekey") is shown in the dialog feed and recorded in the journal.

Special cases:

- a MIFARE Plus card in factory state (SL0) and a DESFire card without the profile application cannot be written directly—run Card Initialization first (the dialog will offer a link);
- for DESFire cards, rotating application keys is performed exclusively via Card Initialization—the source profile field is not shown for them during card reissue;
- a foreign card (matching neither the target nor the source key) is left unmodified.

Card Initialization

Initialization transitions a card from its current (transport, factory, or prior profile) keys to the keys of a target card profile from the library—without creating an identifier in the system. This is used both to prepare blank cards for issuance and to rotate keys on already issued cards (re-key). For DESFire cards, initialization creates the profile's application on the card (Application AID, code file, and keys)—codes can be written to the card only after this step. Initialization zeroes out the profile data area: if a code was previously written on the card, it is erased. Re-initializing a card that already uses the target profile keys (when the source and target profiles are the same) is the standard way to wipe card data without changing keys. The **Initialization** button in the **Identifiers** panel requires the **Card initialization** permission (Section 14.2) and an assigned Reader Workstation (Section 14.8).

1. Click **Initialization**.
2. Select **Source profile (current card keys — optional)** — for a new MIFARE card, select a transport profile from the library (Section 14.9); transport profiles are badged and listed first. For a DESFire target profile, this field is labeled **Source profile (application re-key — optional)**: specify it only if the card already carries an application under another profile's keys (application key rotation); the Application AIDs of both profiles must match.
3. Select **Target profile (new sector keys)** — a MIFARE Classic/Plus or DESFire profile (DESFire profiles are badged in the list, and for them the field is labeled **Target profile (DESFire application and keys)**).
4. For a MIFARE Plus target profile, select **Target Plus security level**: **SL3 (AES) — target mode, without Classic compatibility** or **SL1 — Classic mode, upgrade to SL3 possible later**. Personalization from SL0 requires Plus admin keys in the profile (Section 14.9).
5. Place the card on the desktop reader; optionally probe its state first by clicking **Check card**. If the probe detects an incomplete DESFire application (left over from an interrupted previous initialization), queuing the job requires explicitly checking the resume confirmation box in the dialog.
6. Check the risk acknowledgment box and click **Initialize**. Keep the card on the reader until completion.

For a DESFire target profile, the dialog also provides the **Format card** button—a full factory reset of the card (`FormatPICC`) with confirmation (erases all applications and keys; this action is irreversible). Formatting is possible only if the card's PICC master key is at its factory default; after formatting, run initialization.

> **Warning.** Card initialization and MIFARE Plus security level transitions are irreversible: re-keying overwrites sector keys, switching to SL3 permanently disables MIFARE Classic backward compatibility, and the Plus security level can only be upgraded, never downgraded. If the card is removed from the reader mid-write, the sector may remain on unknown keys, making recovery outside the system impossible.

Lifecycle Constraints

- The code of an existing identifier cannot be edited. To replace a code, use the **Reissue** operation.

- Letter case in the code is ignored: codes are converted to lowercase when saved, and case is disregarded when a card is presented.
- The **Revoked** state is irreversible. A revoked identifier cannot be reactivated, assigned, or reissued.
- The code of a revoked identifier remains reserved in the database, and its event history is preserved.
- An identifier in the **Lost** state is automatically unbound from its person. It can be assigned again only after changing its state.
- The **Lost** and **Revoked** states serve different purposes: **Lost** is set when a card goes missing, whereas **Revoked** is set when a card is replaced via reissue or permanently decommissioned.

Identifier Templates

Templates are configured in the **Identifier templates** section of the tool panel. A template defines the identifier kind and the rule for calculating its validity period:

- **Unlimited**;
- **For 3 months**;
- **For 6 months**;
- **For 1 year**;
- **Until date**.

For relative periods, validity is calculated from the identifier's creation date. For **Until date**, the fixed expiration date specified in the template is used.

12.7. Events Panel

The **Events** panel displays audit events for persons, identifiers, and passes alongside access events—so you can see both "who modified the pass" and "how it was used at an access point". The **Selected** toggle narrows the feed to the currently selected record—a pass, a person, or an open identifier—combining its audit trail and passage history in a single stream. Techniques for working with the event feed are described in Chapter 8.

12.8. Custom Personnel Fields

The **Custom personnel fields** section of the tool panel configures the structure of the person card:

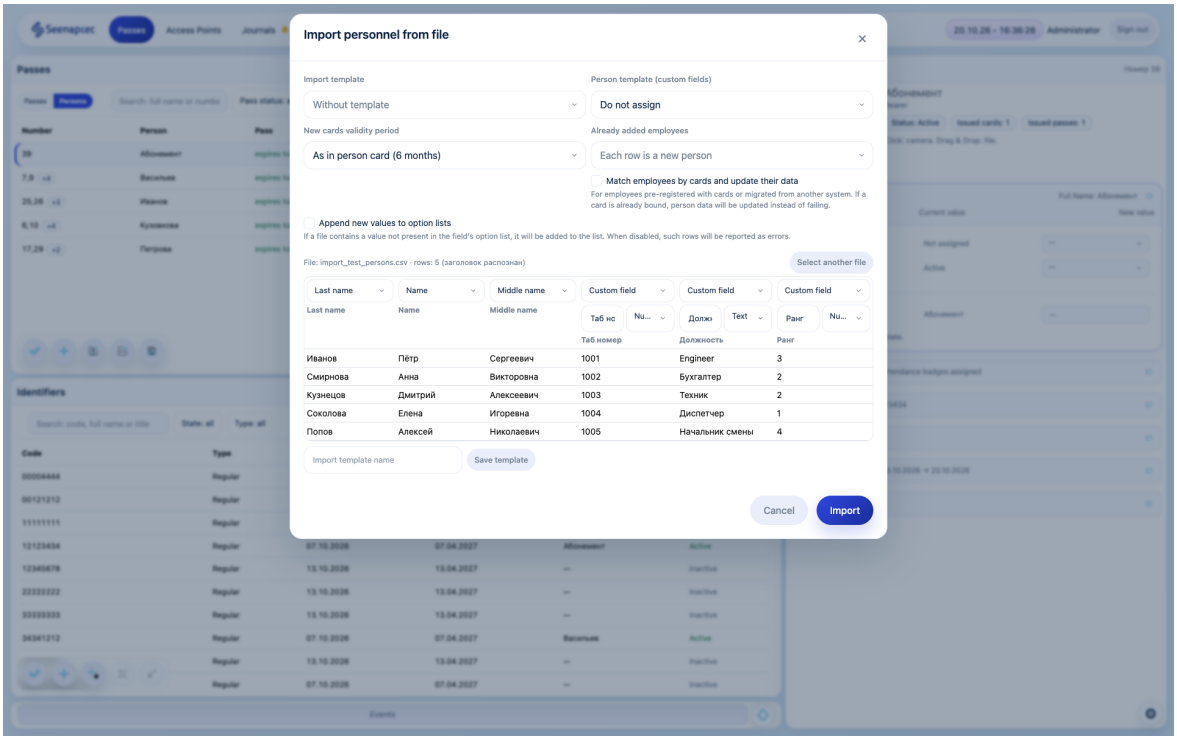
- **Custom field types** — field definitions: name, category (**Personal data**, **Organizational data**, **Contacts**, **Technical metadata**, **Status**, **Other**), value type (**Row** / string, **Integer**, **Boolean**, **Enumeration**), and the **required** and **unique** flags;
- **Personnel field templates** — ordered sets of custom fields; a template is assigned to a person in their card. Fields that were filled in outside the currently assigned template are preserved and displayed in a separate block.

Custom fields power the **By fields** search in the persons list (Section 12.2).

> **For those familiar with traditional ACS.** Custom fields are not free-form "notes columns", but a strongly typed, system-wide directory: the "Department" field is defined once across the entire system rather than as an ad-hoc text string in each card. The **Enumeration** type with a predefined option list prevents data inconsistencies ("IT", "I.T.", "it dept") and makes fields suitable for exact **By fields** queries and reliable person matching during CSV import (Section 12.9). Marking a field as **unique** turns it into an external key—such as an employee ID used to synchronize personnel with an HR system.

12.9. Importing Personnel from a File

The **Import personnel from file** button in the **Passes** panel opens a wizard for bulk-loading people from a CSV file—used during initial system commissioning, scheduled synchronization with HR exports, or migration from another ACS. The import creates and updates persons, their custom fields, and their cards; it does not issue passes—passes are issued afterward through standard or bulk template issuance.



Column mapping. The wizard displays a preview table of the CSV file; above each column, you select its role: **Do not import**, **Full name**, **Last name**, **Name** (First name), **Middle name**, **Cards**, or **Custom field**. For a **Custom field** column, you specify the field name and value type; for enumerations, the list of allowed options is collected from the file data. The wizard automatically suggests standard column roles based on column headers; multiple rows belonging to the same person (for example, one row per card) are merged automatically.

Settings. In addition to column roles, you configure:

- **Person template (custom fields)** — which personnel field template to assign to newly created persons;
- **New cards validity period** — the identifier template applied to cards imported from the file (**As in person card (6 months)** by default);
- **Already added employees** — deduplication and update key: **Each row is a new person** or **Match by field '…' and update** — an external identifier (a unique custom field such as an employee ID) used to update existing persons rather than duplicating them: their full name and custom fields are updated, and any missing cards are added;
- **Match employees by cards and update their data** — an additional matching key for people pre-registered with cards or migrated from another system: if a card from the file is already bound to an employee, the row updates that employee's record;
- **Append new values to option lists** — allows enumeration fields to be expanded with new option values encountered in the file (otherwise, a row containing an unknown enumeration value is rejected with an error).

Import results. Each row of the file is processed independently. An error in one row—such as a card code already in use, a unique field conflict, or an invalid enumeration value—does not halt the import of the remaining rows. Upon completion, the wizard displays a summary showing the number of created, updated, skipped, and failed records, along with an explanation for each failed row. In the journal, the import is recorded as a single summary audit event containing the input row count and final outcome metrics.

Import templates. A configured mapping—column roles, name format, deduplication key, templates, and flags—can be saved as a named **Import template** and applied in one click via the **Import template** field during subsequent imports. Import templates are shared across all operators, reducing routine monthly HR synchronization to "select file → select template → apply".

Permissions: importing requires permission to add persons (`iam.persons.add`); mapping **Cards** columns requires permission to issue identifiers; update/merge modes require `full` permissions on persons; creating or expanding custom fields requires permissions on templates. Unavailable modes are disabled by the wizard with an explanation.

12.10. Auto-Cleanup

The **Auto-cleanup** section of the tool panel manages automated data lifecycle rules:

- **Delete expired passes** + **Retain after expiration (days)** — expired passes (including pass requests) are automatically deleted once the retention period elapses;
- **Delete persons without passes** + **Person inactivity period (days)** — persons who have had no active passes for the specified number of days are automatically deleted along with their identifiers.

Persons in the **Blacklisted** state are never deleted by auto-cleanup. Every automatic deletion is recorded in the journal as individual audit events, and the history of deleted records remains readable (Chapter 13).

Chapter 13. Journals

The **Journals** workspace is the workstation for auditors and security operators: historical event analysis, incident investigations, alarms, archiving, and journal integrity verification. For live monitoring of a specific area, each workspace includes an **Events** panel (Chapter 8); **Journals** is the dedicated workspace for working with the complete system history.

13.1. Journal Architecture

The journal is a unified, append-only stream of events across the entire system: stored records cannot be modified or deleted (except through scheduled or manual archiving, see Section 13.6). Every event is classified along three axes:

- **category** — **Access** (access decisions and passage episodes), **Audit** (Configuration / Maintenance / Security changes made to data and settings by operators and services), **Technical health** (**Equipment** and service status), as well as **FAS** (Fire Alarm System events for panels, zones, loops, and detectors) and **Alarms**;
- **domain** — **Personnel & Passes**, **Access objects**, **Equipment**, **FAS**, **Security**, **Presence tracking**;
- **severity** — **Info**, **Attention** (Notice), **Warning**, **Critical**.

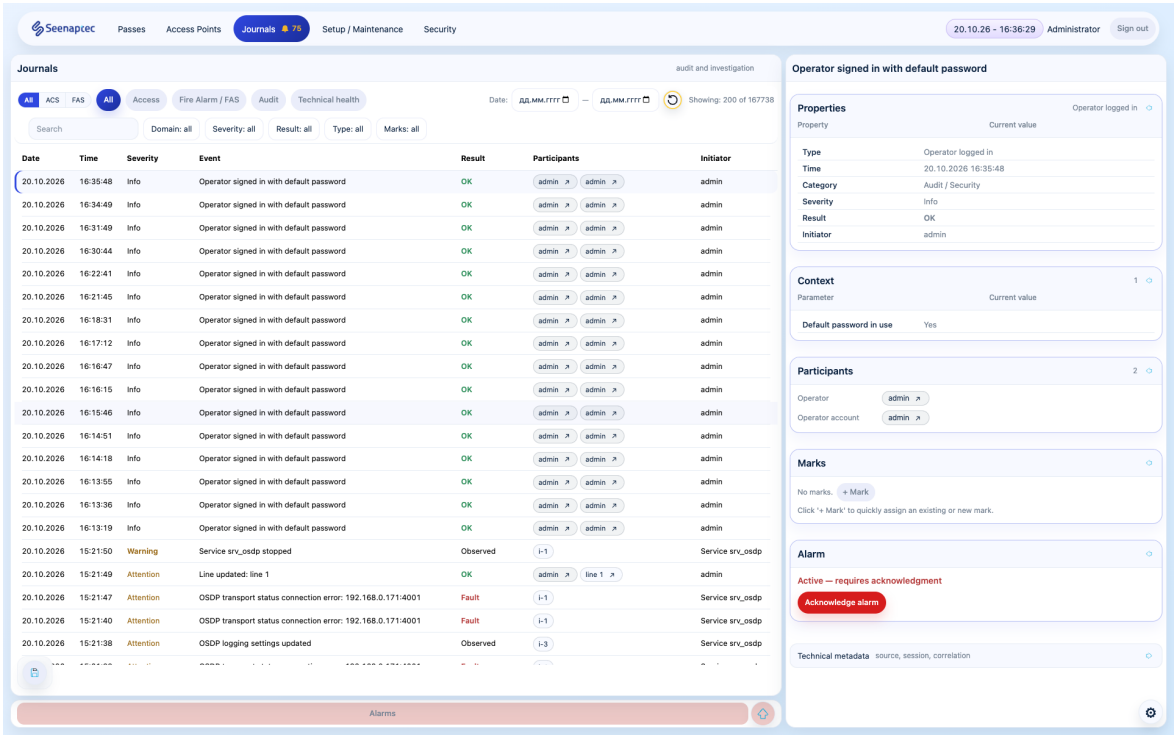
Every mutating operation in the system is required to leave an audit event—this is a core product invariant. Audit events record a **Before / After** table of modified field values, while access events record the participants and the decision reason.

Events that occur on a field device while communication with the server is lost—such as emergency offline unlock activations (Section 11.3)—are never lost: the device maintains its own local offline log, and once communication is restored, the system imports those events into the main journal stream with their actual device timestamps and built-in deduplication.

The only mutable layer on top of stored records is operator **marks** (Section 13.3); adding or removing a mark is itself recorded in the audit journal. Marks allow operators to tag and quickly retrieve events of interest during investigations.

References to deleted entities. Person display names are captured at the time of the event, and numeric system IDs of entities are never reused—ensuring that the event history remains accurate and readable even after entities are deleted from the system. The badge of a deleted participant is rendered in a muted color with a `(deleted)` suffix in its tooltip; navigating to the entity's card is unavailable, but filtering the journal history by that participant continues to work.

13.2. Journals Panel: Searching History



The top bar of the panel contains the category switcher: **All** | **Access** | **Audit** | **Technical health** (along with subsystem/category filters covering **Alarms**, **Equipment**, **Configuration**, **Maintenance**, **Security**, and **FAS**). In addition, while the **Journals** workspace is open, the top application bar displays the three-state subsystem pill **`All | ACS | FAS`**, allowing you to filter both the journal feed and the **Alarms** panel to ACS-only or FAS-only events with a single click. Below it are the filter controls:

- **period** — **Date: from — to** fields (inclusive); the red × button resets the date range;
- **Domain**, **Severity**, **Result** (**Granted**, **Denied**, **Registered**, **OK**, **Fault** / **Failed**, **Observed**), and **Type** — multi-select dropdowns; the list of available event types narrows automatically based on the selected category and domain;
- **Search** — substring search across event descriptions and participant labels;
- **Mark** — filters events carrying a specific operator mark.

The list displays the following columns: **Date**, **Time**, **Severity**, **Event**, **Result**, **Participants**, and **Initiator** (**Actor**). Historical records load incrementally as you scroll; the **Showing: N of M** counter indicates the number of loaded rows versus total matching events.

Entity history (object filter). Clicking any participant badge adds an entity filter—narrowing the feed to all events involving that specific object (including deleted entities). Active entity filters are displayed in the **Entity history:** bar with a **Reset** button. This is the primary investigation workflow: viewing "everything that happened with this door / this pass / this operator".

Presence tracking. Events in the **Presence tracking** domain record the closing of a presence interval: person, tracking zone, duration, and close reason (passage, timeout, operator reset, or cabinet session end). To filter these records, select the **Presence tracking** domain and, if needed, the **Presence interval closed** event type.

Navigating to a participant and returning. In addition to filtering, each participant badge—both in the event list and in the **Participants** section of the event card—features an **Open object** arrow button: clicking it opens the entity itself in its native workspace (a person, pass, or identifier in **Passes**; an access object or fire zone `D`, `A`, `P` in **Access objects**; a device, line, server, or FAS panel/loop/detector in **Configuration / Maintenance**). Clicking the main body of the badge still adds an entity filter. After navigating away, a back button appears in the header of the target workspace's main panel, returning you directly to **Journals** with the same event selected (Section 8.10). Deleted participants do not have the navigation arrow (since the target card no longer exists), but history filtering by their badge remains available.

Lower "Events" panel operation with the FAS subsystem. In the **Setup / Maintenance** and **Access Objects** workspaces, the lower **Events** panel automatically follows the **`ACS | FAS`** switcher: in `FAS` mode, the feed displays fire alarm and automation events, and enabling the **Selected** toggle filters the history to the currently selected fire zone (`D`, `A`, `P`), UniPOS panel, signaling

loop, or addressable FAS device.

Exporting Results to CSV

The **Export** button at the bottom left of the panel exports the current query results to a CSV file (and JSON structures in archive/diagnostic workflows)—including all events matching the active panel filters, not just the rows currently scrolled into view. In archive viewing mode (Section 13.7), the export button is disabled: only the live database can be exported.

The **Export journal to CSV** dialog displays the total number of matching events and the active filter summary. You can customize the exported columns and formatting:

- **base columns** are always included: **Date/time**, **Type**, **Category**, **Severity**, **Result**, **Actor**, **Summary**;
- **participants** — one column per participant type (person, pass, access object, device, etc.); only participant types relevant to the active filter are offered, all enabled by default—use the **All** and **Clear** buttons to toggle the entire list;
- **context fields** — one column per field, grouped by topic (**Access**, **Authentication**, etc.); disabled by default, leaving cells empty for events that do not carry a given field;
- **delimiter — Semicolon (;)** or **Comma (,)**;
- **values — Localized (as in journal)** (human-readable labels as displayed in the UI feed) or **Raw values** (**Raw database codes**, intended for automated machine processing).

Clicking **Download CSV** downloads a UTF-8 file named `journal-export-YYYYMMDD-HHMMSS.csv`. Two constraints are noted directly in the dialog:

- the operator mark filter is not applied to the export—the file contains all matching events regardless of marks;
- at most 50,000 events are exported per file; if the result set is truncated, the final row of the file contains a truncation marker indicating the total number of matched events and advising you to narrow the filter for a complete export.

Every export operation is recorded in the audit journal as a **Journal events exported** event containing the filter summary and the number of exported rows—making it clear who exported what data.

13.3. Event Card

To open an event card, select its row in the journal. Card sections can be collapsed and expanded by clicking their headers. The system remembers which sections are open on the current workstation and preserves that layout when you inspect other events.

Depending on the event type, the card contains the following sections:

- **Properties** — event type, date and time, category, severity, result, and initiator. The section header displays the event type name.
- **Participants** — roles and linked entities: access object, person, pass, identifier, device, line, server, or operator. Click a participant badge to filter the journal by that entity, or click the arrow on the badge to open its card.
- **Changes** — a **Field / Before / After** table showing modified values. Displayed for audit events.
- **Links** — added and removed associations between entities, such as changes to the set of access objects in a pass.
- **Context** — additional event parameters: identification mode, passage direction, decision reason, and other details. Access object components are shown by their assigned names, such as “Entry reader” or “Exit button”.
- **Marks** — operator tags used to group events during an investigation. To add a mark, type it into the **New mark** field. To remove a mark, click the delete button on its badge. Marks are visible to all operators, and adding or removing them is logged in the audit journal.
- **Alarm** — the state of an alarm event. For an active alarm, the status **Active — requires acknowledgment** and the **Acknowledge alarm** button are displayed. Once acknowledged, the card shows the acknowledging operator, date, time, and comment.
- **Technical metadata** — event ID, server, source service, and correlation IDs. Collapsed by default.

When viewing an event from another workspace’s **Events** panel, click **Open in Journals** to open the same event in the **Journals** workspace and continue your investigation with full filtering tools.

13.4. Alarms

An alarm is an event that requires operator attention and acknowledgment. Examples of alarms include an access breach, device tamper sensor activation, journal integrity violation, secret retrieval failure, and access under duress. The complete list

of alarm events is defined by the system event registry.

Alarm Indicator

Whenever there are active unacknowledged alarms in the system, the **Journals** workspace button in the top navigation bar displays a bell icon with a counter showing the number of active alarms. The arrival of a new alarm is accompanied by an audible alert.

Click the bell icon to open the **Alarms** panel from any workspace.

Alarms Panel

The **Alarms** panel lists only active unacknowledged alarms. The panel header shows their total count, and the table displays the following columns:

- **Date**;
- **Time**;
- **Severity**;
- **Event**;
- **Participants**.

Select a row to open the alarm acknowledgment card. Clicking the arrow on a participant badge opens the linked entity in its corresponding workspace.

13.5. System Journal Integrity Verification

Records in the system journal are linked into a cryptographic hash chain. Each record contains a cryptographic digest of its own payload combined with the digest of the preceding record. This guarantees that any modification of event contents, insertion, or deletion of records will be detected.

Periodically, the journal is partitioned into "sealed" segments. The cryptographic anchor digests of sealed segments are additionally stored in the system configuration database, ensuring that any modification, insertion, deletion, or even a complete rewrite of the journal's tail is detected during verification.

Verification Methods

Journal chain integrity is verified:

- automatically — at system server startup and periodically during runtime;
- manually — in the **Journal integrity** section of the tool panel;
- when opening archive files (Section 13.6).

The **Journal integrity** section displays summary metrics:

- total number of events in the live journal;
- number of sealed segments.

To run a manual check, click **Check / Verify** (**Verify chain**). The system verifies the unsealed tail of the journal—from the last sealed segment up to the most recently recorded event.

If verification succeeds, the system displays:

Integrity verified: events #A–#B (N pcs.)

If a violation is detected, the system reports the violation kind and the ID of the first corrupted record.

Possible integrity violations:

- modification of an event row's contents (`row`);
- broken hash chain due to record insertion or deletion (`chain`);
- corrupted sealed segment (`segment`);
- mismatch between segment anchor digests and the anchors stored in the configuration database (`anchor`).

When an integrity violation is detected, the system:

- raises a **Journal integrity violation** alarm;
- records an audit event with the verification result;
- continues recording new incoming events in the journal.

13.6. Archiving

Archiving moves older sealed journal segments into standalone archive files and frees space in the live database without breaking the cryptographic hash chain. It is configured in the **Archiving** section of the tool panel:

- **Auto-archiving** — enables scheduled execution, with configurable **Frequency** (days or months) and **Start time**;
- **Archive folder** — the target directory on the server filesystem;
- **Password** — when a password secret (from the **Security** secrets vault) is selected, archive files are encrypted; if set to **Unencrypted**, files are exported in plaintext. The same password secret can be shared between journal archiving and system backups (Section 10.8).

Clicking **Archive now** starts an immediate manual archive run. Only sealed segments are archived; once a segment is successfully written to disk, its records are purged from the live database strictly in chronological order. A segment containing an unacknowledged alarm is exported to an archive file but is retained in the live database until the alarm is acknowledged. The outcome is reported in a status message ("exported X segments, purged Y from database") and logged as journal events.

Each archive file contains an unencrypted header (manifest with the event range and cryptographic root hashes) followed by the full segment records; encryption protects the event payload while leaving the header readable, allowing the server to inventory the archive folder without requiring the password.

13.7. Viewing Archives

The **Archive** section of the tool panel opens archived journal files in read-only mode:

1. In the file list (showing date range, event count, file size, and encryption badge), select one or more **contiguous** segments. For encrypted files, the system automatically uses the archive password configured in settings if available; otherwise, an **Archive password** prompt appears—using the same interface as backup restoration (Section 10.8). An external archive file can also be opened from your computer via the **Upload file…** button.
2. Click **Open archive**. The system verifies the cryptographic integrity of each file and the continuity between adjacent segments, then mounts a temporary read-only database.

The workspace switches into archive mode: you use the same event feed, filters, and event cards, while an **ARCHIVE** banner at the top displays the loaded date/event range and the integrity verification result (integrity verified or violated, with the count of detected violations—violations do not block viewing). In archive mode, alarm acknowledgment and mark editing are disabled—marks stored inside the archive are displayed as historical read-only tags. Clicking **Return to live journal** closes the archive viewing session.

13.8. Reports

The **Report generation** section of the tool panel builds analytical reports from journal data. The current version supports the **presence tracking report**: total time spent by persons inside the zone of a selected presence tracking badge (Section 11.11) over a specified period. The data source is **Presence interval closed** events in the **Presence tracking** domain (Section 13.2); currently open intervals are not included in the report.

Report Definitions

Reports are stored in the section as a list of named definitions. A report definition specifies:

- the report name;
- the presence tracking badge (zone);
- the reporting period: **Today**, **Yesterday**, **This week**, **Last week**, **This month**, **Last month**, or a custom date range;
- selected persons (all persons by default) and an optional filter by custom personnel fields;
- custom personnel fields to include as additional columns in the exported report;
- the CSV delimiter: `;` or `,`.

Click **Save** to save the definition, or **Save and build** to save the definition and immediately generate the report file.

Building a Report

When running a report, you can temporarily override the tracking badge, period, or person in the **Run parameters** block—without modifying the saved report definition. This makes it easy to generate the same report "for last month" or "for a single employee" on the fly. Relative periods are evaluated using the server's clock at build time, with weeks starting on Monday.

Each row in the generated report represents the daily total for a `date + person` pair:

- **First entry** — timestamp of the first entry into the zone on that day;
- **Last exit** — timestamp of the last exit among reliable intervals on that day;
- **Zone presence time** — the sum of durations of reliable intervals for that day (not the elapsed time between first entry and last exit: breaks spent outside the zone are excluded);
- **Unreliable** — the count of intervals closed by timeout (Section 11.11); timed-out intervals are excluded from **Zone presence time**, and this column is left blank if there are none.

An interval is attributed to the calendar date of its entry and is not split at midnight: an overnight shift is recorded in full under the date it started.

Report File

Building a report produces a UTF-8 CSV file with a BOM (ensuring proper character rendering in spreadsheet applications), formatted with the delimiter specified in the definition. Only the most recently built file is retained for each report definition: building the report again overwrites the previous file. If no matching data exists for the selected period, the previously generated file is kept intact. Click **Download** to save the file locally, or **Delete file** to remove the generated file without deleting the report definition.

Creating, modifying, and deleting report definitions, as well as every report build, are recorded as audit events in the journal—ensuring that every export of employee working time records is auditable regardless of who subsequently downloads the file.

Chapter 14. Security

The **Security** workspace governs access to the system itself: operator accounts, roles and RBAC permissions, active sessions, the secrets vault (**Keys**), security settings, and platform audit logs. It also configures the desktop card subsystem: desktop readers (Reader Workstations) and the library of card profiles. The left panels of the workspace are **Operators** (with the **Operators / Reader Workstations** switcher), **Events**, and **Sessions**; the tool panel contains the **Roles**, **Secrets**, **Security settings**, and **Card profiles** sections.

14.1. Permission Model (RBAC)

An operator's permissions are determined by their assigned role (plus any individual permission overrides granted on top of the role). A permission defines the access level (`none` / `read` / `add` / `full`) for an area within one of the system's RBAC domains:

Domain	Scope
Equipment	Lines, devices, firmware, backups, HTTPS/TLS, reader programming cards, maintenance operations
Access objects	Access points, rules, operational control, live access monitoring
Persons and passes	Persons, identifiers, passes, card initialization, templates and directories
Security	Operators, roles, secrets, Reader Workstations, card profiles, system/platform audit
Journal	Event journal and reports, alarms, journal integrity, journal archives

Permission levels are cumulative—each higher level includes all capabilities of the preceding ones: **None** (– / `none`) → **Read** (`read`) → **Add** (`add`) → **Full** (`full`). In the **Passes and identifiers** area (`iam.passes`), there is an intermediate level between **Read** and **Add**: **Issue by template** (`issue_tpl`), which authorizes issuing passes strictly from templates marked for restricted issuance (Chapter 12).

In the permission matrix, areas within each domain are divided into **base areas** (the top-level entry of the domain: for example, **Lines and devices** in **Equipment**, **Access points: configuration** in **Access objects**, **Persons** and **Passes and identifiers** in **Persons and passes**, **Operators and sessions** and **Reader workstations** in **Security**, and **Journal and reports** in **Journal**)

and **dependent areas** (nested rows below the base areas). Selecting any permission level on a dependent area in the matrix automatically enables **Read** (`read`) on the base area of that domain; clearing all base areas of a domain (– / `none`) automatically resets its dependent areas as well.

Subsystem access (permission ceilings for `ACS` and `FAS`). Above the domain matrix in both the role editor and operator permission overrides is the **Subsystem access** block containing two top-level scope controls:

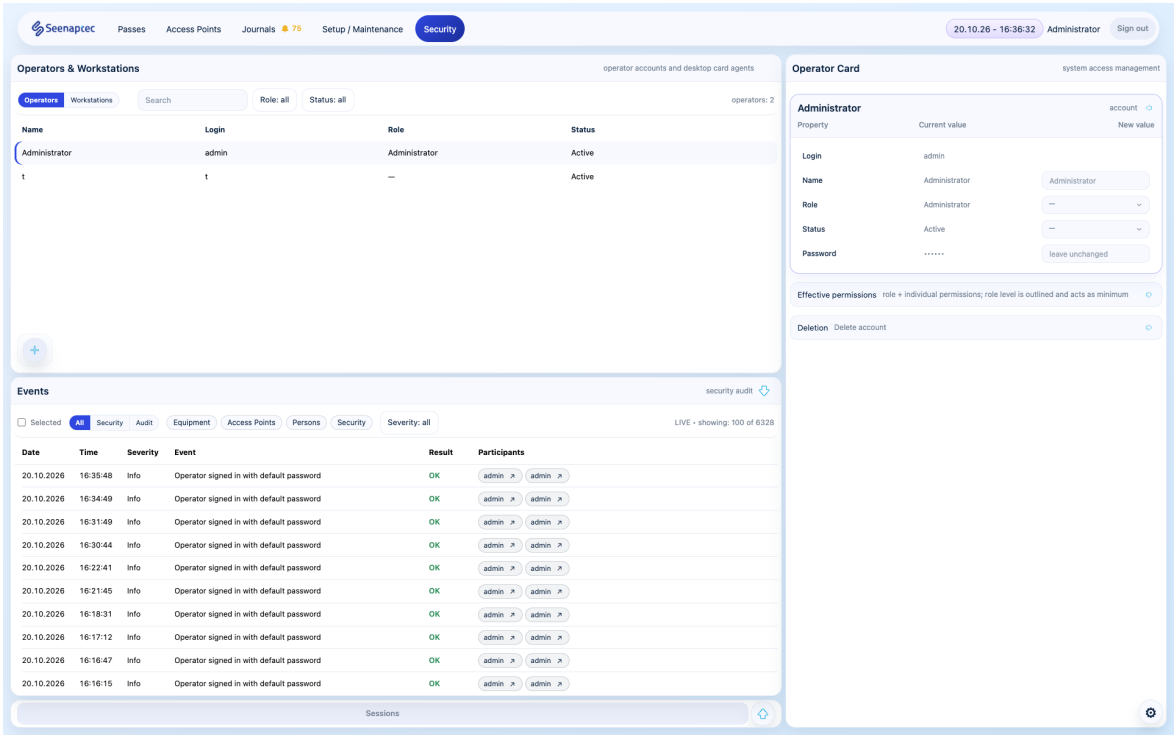
- **Access Control System (ACS)** (`domain.skud`);
- **Fire Alarm System (FAS)** (`domain.ops`).

Each subsystem has a permission ceiling (**None** / **Read** / **Add** / **Full**) that caps the maximum effective permission level the role or operator can exercise within that subsystem (for example, if `domain.ops` is set to **Read**, the operator can view FAS panels and fire zones but cannot issue activation or bypass commands or modify panel settings, even if they hold **Full** permissions in the **Equipment** and **Access objects** domains). If one of the subsystems is set to **None** (–), that subsystem is completely hidden from the operator, and the **`ACS | FAS`** switcher in the top bar is automatically hidden, keeping the workspace strictly within the permitted subsystem.

Anti-escalation. An operator can never grant others higher permissions than they hold themselves: when editing roles or operator overrides, levels above the operator's own permissions are disabled (with the tooltip **Above your permission level in this area**), and the server enforces the same invariant. The built-in **Administrator** (**Super Administrator**) role cannot be modified or deleted; only an administrator can assign it or modify administrator accounts.

The user interface adapts to effective permissions automatically: workspaces, left panels, and tool panel sections without `read` access are hidden, while unavailable action buttons are disabled with the tooltip **Insufficient permissions**.

14.2. Operators Panel



The list of operator accounts displays the columns **Name**, **Login**, **Role**, and **Status** (**Active** / **Disabled**); above the list are a search box and multi-select checkbox filters for role and status (selections are applied by clicking the confirmation button, just as in the **Access** panel filters). At the bottom of the panel is a single + button; bulk operations are not used for operators. The **Operators / Reader Workstations** switcher in the panel header switches the panel to desktop card reader management (Section 14.8).

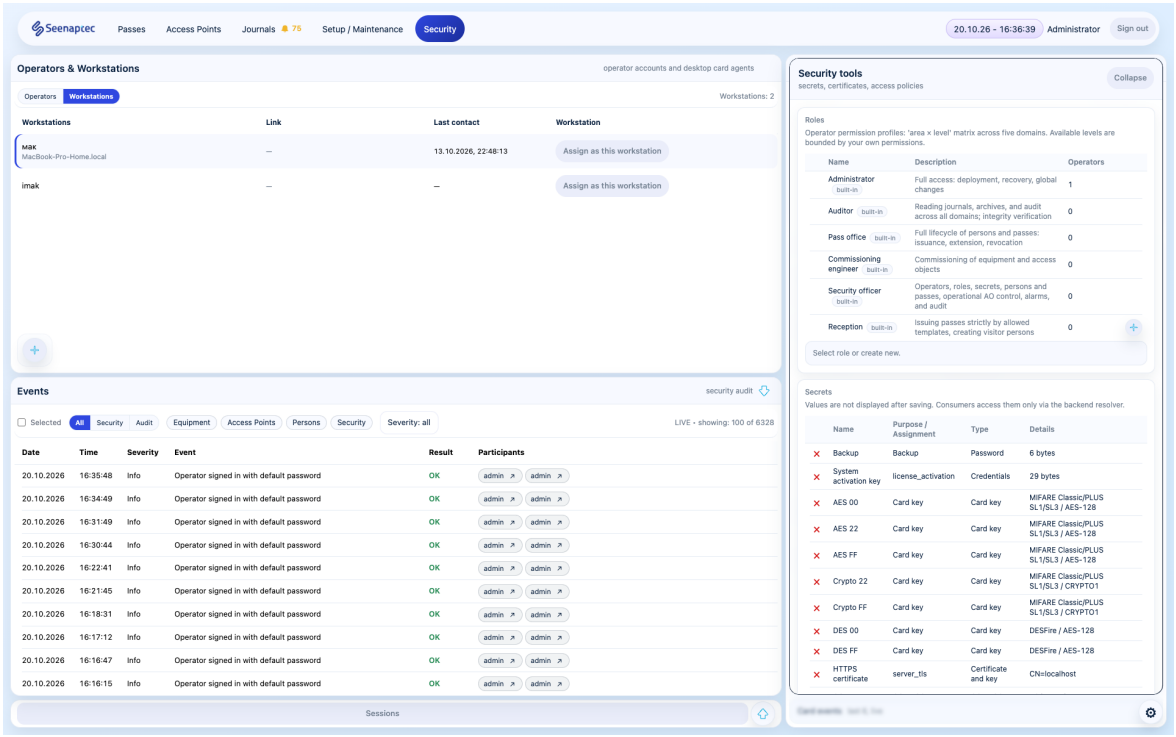
Creating an operator. Clicking + (**Add operator**) opens the **New operator** dialog: **Login**, **Name** (defaults to the login), **Role**, and **Password**. Password requirements are governed by the system security settings (Section 14.6).

Operator card. The first section contains account properties: login (immutable), display name, role, status, and password change (the current password is never shown; leaving the field blank keeps the existing password). A password set for an operator by another operator is treated as administrator-assigned: when the **Require changing administrator-assigned password** policy is enabled (Section 14.6), the account holder will be required to set a new personal password upon their next login. The second section is **Effective permissions**: the resulting permission matrix combining the assigned role and individual overrides. The permission level granted by the role is marked in each row with an asterisk to the left of the label—this represents both the inherited value and the minimum floor: levels below the role level cannot be selected (tooltip **Below role level**), as individual overrides can only elevate permissions above the role. Selecting the starred level resets that area back to the role's default (removing the individual elevation); the server stores only overrides that exceed the role, so any future changes to the role immediately take effect in the operator's effective permissions.

Disabling and deleting. To temporarily suspend an operator's access, set their **Status** to **Disabled**. Permanent deletion is performed from the bottom of the operator card: the collapsed **Deletion** section explains the consequences—the operator's active sessions are terminated immediately while their audit history in the journal is preserved—and clicking the delete button requires confirmation in the right panel. Safeguards: you cannot delete your own account (the button is disabled with an explanation, and the server rejects such requests), and you cannot delete the last active operator holding `full` permissions on operator management—preventing the system from being left without account administration.

All changes are recorded in the audit journal: creating, modifying, and deleting operators, as well as permission changes.

14.3. Roles



The **Roles** section of the tool panel manages permission profiles: an `area × level` matrix across the RBAC domains.

Preconfigured roles (badged as **built-in**):

Role	Purpose
Administrator (**Super Administrator**)	Full unrestricted access (`full` across all domains): deployment, backup/restore, global configuration
Security officer (**Security Officer**)	Operators, roles, secrets, persons and passes, operational control of access points, alarms, and system audit
Pass office (**Pass Office**)	Full lifecycle of persons, identifiers, and passes: issuance, extension, revocation
Reception (**Operator**)	Template-restricted pass issuance (**Issue by template**), visitor registration, and operational monitoring

Role	Purpose
Auditor (**Observer**)	Read-only access (`read`) to the event journal, archives, reports, and audit logs across domains; integrity checks
Commissioning engineer (**Engineer**)	Hardware commissioning, lines, devices, firmware, and access point configuration

Custom roles are created by clicking **Create role** (name, description, **Subsystem access** ceilings for `ACS` and `FAS`, and the permission level matrix across the five domains) and edited using the same interface; the role list shows how many operators are currently assigned to each role. Built-in roles cannot be deleted, and a custom role cannot be deleted while it is assigned to one or more operators.

14.4. Sessions Panel

The **Sessions** panel displays a live list of active operator sessions with the columns **Operator** (your own session is marked `(you)`), **IP**, **Browser**, and **Activity**; the list refreshes automatically, and the header counter shows `sessions: N`.

Sessions use a sliding inactivity window: every request from the operator extends the session, and if the session remains idle longer than the configured **Session idle timeout (min)** (Section 14.6), it expires and requires signing in again. An administrator can forcibly terminate another operator's session by clicking **Terminate session**—revocation is logged in the audit journal as an **Operator session revoked** event; sessions belonging to administrators can only be terminated by an administrator.

14.5. Secrets (Keys)

The **Secrets** section of the tool panel is a write-only encrypted vault for sensitive cryptographic material (**Keys** and credentials) that the system uses internally but never displays back: backup encryption passwords, journal archive passwords, OSDP Secure Channel keys, MIFARE/DESFire card keys, and HTTPS/TLS certificates.

Each secret has a name, a purpose (such as `Backup`, `OSDP SCBK`, `Card key`, or `HTTPS/TLS`), and a material type (**Password**, **Symmetric key**, **Certificate**, etc.). Secret material is entered once during creation or rotation: once saved, it is encrypted at rest and never shown on any screen—you can replace the value, but you cannot read it back. Subsystems (backup, HTTPS, OSDP lines, card profiles) reference secrets by ID.

Creating, rotating, and deleting secrets are logged in the audit journal—without exposing secret values.

14.6. Security Settings

The **Security settings** section of the tool panel configures system-wide authentication and session policies:

Parameter	Default	Description
Session idle timeout (min)	720	A session expires after the specified number of idle minutes; active requests extend the window. Valid range: 1 minute to 7 days (10,080 minutes).
Allow empty password	No	Allows operator accounts without a password (intended for test benches; not recommended in production).
Password security policy	**Standard**	**No** (`none`) — no restrictions.**Standard** — at least 6 characters.**Strict** — at least 10 characters, requiring both letters and digits.
Require changing administrator-assigned password	No	An operator whose password was set by someone else must set their own personal password upon next login—until then, access to the system is blocked.

The password policy is enforced when creating an operator or changing a password; changing the policy does not invalidate already existing passwords. Saving security settings is logged in the audit journal as a **Security settings updated** event with **Before / After** values.

Requiring change of an administrator-assigned password. A password is flagged as "assigned" whenever it was not set by the account holder themselves: either an administrator set or reset the operator's password, or the password was created automatically by the system (the default `admin` account with default password `123456` created during initial activation, see

Chapter 6). When this requirement is enabled, such an operator is presented with the **Change Password** screen (current password, new password twice) immediately after signing in and cannot use the system until they set their own password; the new password is validated against the active password security policy and **cannot match the default password (`123456`)**. Changing the password automatically revokes all other active sessions of that operator.

The "password assigned" flag is always tracked regardless of whether the setting is currently enabled—allowing you to enable the policy retroactively: all operators whose passwords were set by an administrator and have not yet been changed by the holder will be prompted with the **Change Password** screen on their next login. Enabling this policy is recommended in all production deployments where accounts are provisioned centrally, as it guarantees that operators' working passwords are known only to themselves. By default, the setting is disabled so as not to obstruct test benches and demonstrations.

Signing in with the default password. Regardless of security settings, every login using the default password `123456` is recorded in the journal as an **alarm** login event flagged with **Default password in use**—raising an alarm on every such login until the password is changed. This acts as a safety net when mandatory password change is disabled but the default password remains in use.

Changing your own password. An operator can change their own password either on the mandatory **Change Password** screen (when triggered) or—if they have access to the **Security** workspace—directly inside their own operator card in the **Operators** panel. A password set by the operator for themselves is not flagged as administrator-assigned and does not trigger another mandatory change prompt.

14.7. Events Panel: Platform Audit

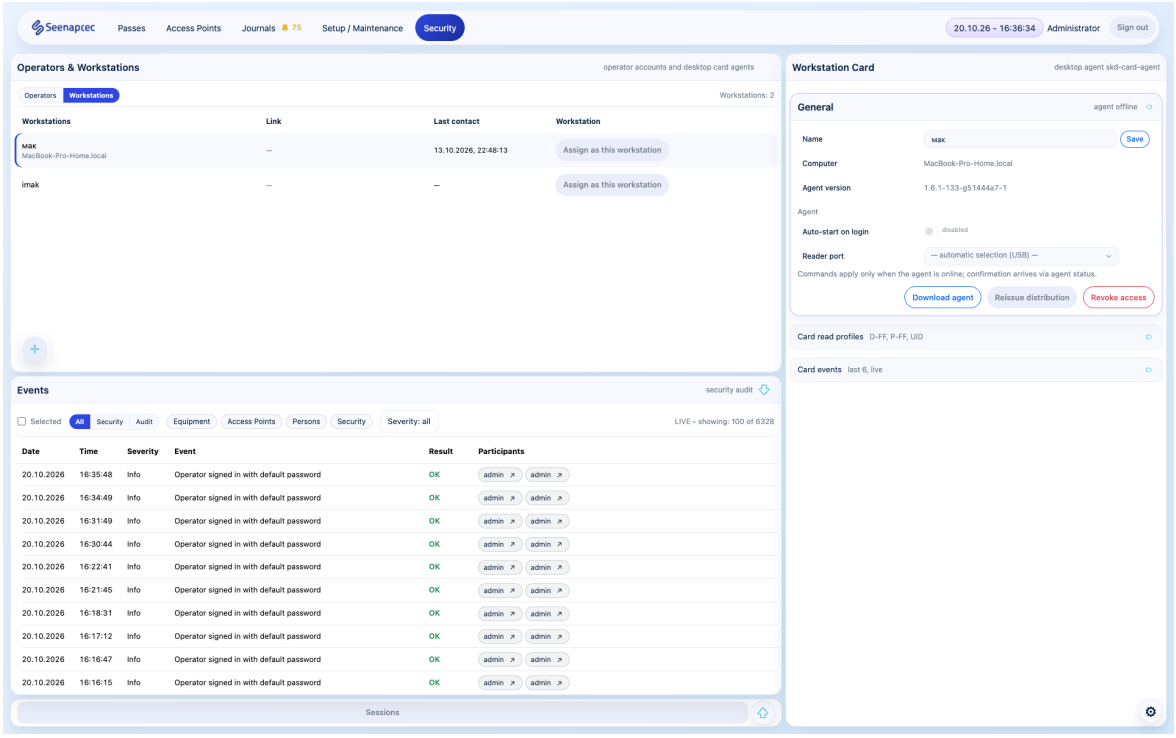
The **Events** panel of the **Security** workspace displays system-wide audit records and platform technical events. The category switcher offers **All** | **Security** (platform technical events: service startups and stops, journal integrity checks) | **Audit**; in **Audit** mode, domain filter buttons are available: **Equipment**, **AO** (access objects), **Persons**, and **Security**.

This feed captures operator logins, failed login attempts, changes to operator accounts, roles, and permissions, session revocations, secret operations, security setting updates, backup restorations, and HTTPS configuration changes. Signing in with the default password is highlighted as an alarm (Section 14.6). The **Selected** toggle narrows the feed to the currently selected operator—showing both actions performed by that operator and changes made to their account.

A specific operator's complete activity history can also be investigated in the **Journals** workspace by clicking the operator's participant badge (Chapter 13).

14.8. Reader Workstations (Desktop Reader / Card Agent)

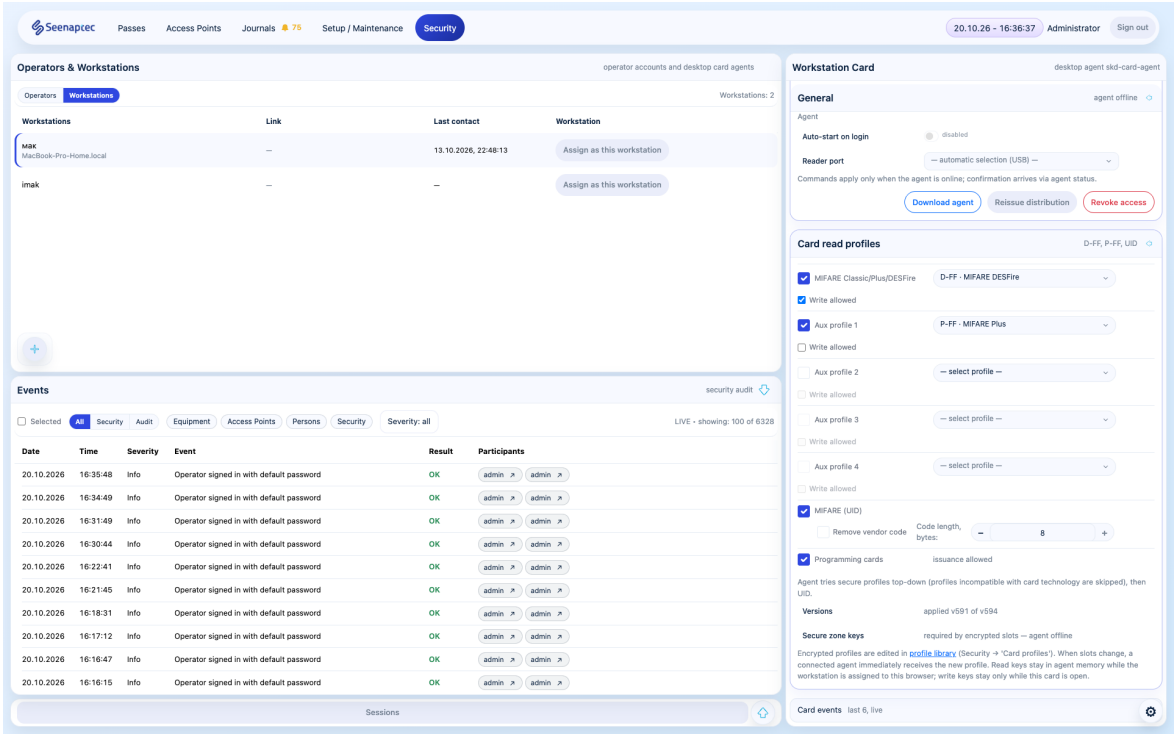
A Reader Workstation (**Desktop reader / Card agent**) is an operator workstation equipped with a USB desktop card reader: a lightweight local card agent service runs on the operator's computer, communicates with the USB reader, and performs reading and cryptographic writing of secure cards—secure card issuance, card reissue, and card initialization (Section 12.6). Reader Workstations are managed in the **Operators** panel switched to **Reader Workstations** mode; this requires the **Security: Reader workstations** (`security.agent_authorize`) permission area.



The Reader Workstations list displays the columns **Workstation** (name and host computer), **Connection**, **Last contact** (timestamp of the last heartbeat/session), and **Workstation** (browser binding). Commissioning workflow:

1. Create a Reader Workstation by clicking + — only a display name is required.
2. Bind the Reader Workstation to your current browser by clicking **Assign as this workstation** in the **Workstation** column. Assigning binds the Reader Workstation to this browser: card enrollment and secure card writing dialogs in **Passes** will use this desktop reader, and downloading the agent installer package is permitted only from the bound browser. The binding is persistent: a workstation token is stored in the browser and the binding record is stored on the server, so it survives server restarts and reconnects automatically without operator intervention. Because the binding is tied to the browser rather than to a specific user account, another operator signing in at the same physical desk does not need to rebind—any operator with card permissions can use the desktop reader immediately upon signing in. The bound entry is badged **This browser's workstation**, and can be unbound via the **Unbind** button. Clearing browser site data erases the local workstation token—if that happens, simply click **Assign as this workstation** again. Checking **Desktop reader only** below the list disables capturing card reads from wall-mounted field readers while a workstation is bound to the browser, preventing passages at doors from accidentally populating input fields on screen.
3. In the Reader Workstation card, click **Download agent** — the server builds a personalized installer bundle containing a one-time activation code. The activation code remains valid for one hour: install and start the card agent on the operator's computer within that window. The bundled configuration includes the SHA-256 fingerprint (pin) of the server's TLS certificate, allowing the agent to trust a self-signed server certificate out of the box without manual OS certificate store configuration. If the server's HTTPS certificate is later reissued or replaced (**HTTPS access**, see Section 10.8), download the agent bundle again and update the agent configuration: an agent with a stale certificate fingerprint will refuse to connect and log a certificate fingerprint mismatch error.

Reader Workstation card. The **General** section shows the workstation name (with a **Save** button), host computer name, and agent version; the **Agent** block contains the **Auto-start on login** toggle (registers the agent in the operating system's login startup items) and the **Reader port** selector (`— automatic selection (USB) —` or a specific port reported by the agent; the chosen port is saved in the agent's local configuration and persists across restarts). Commands can be applied only while the agent is online, and completion is confirmed via the agent's live status report. Action buttons:



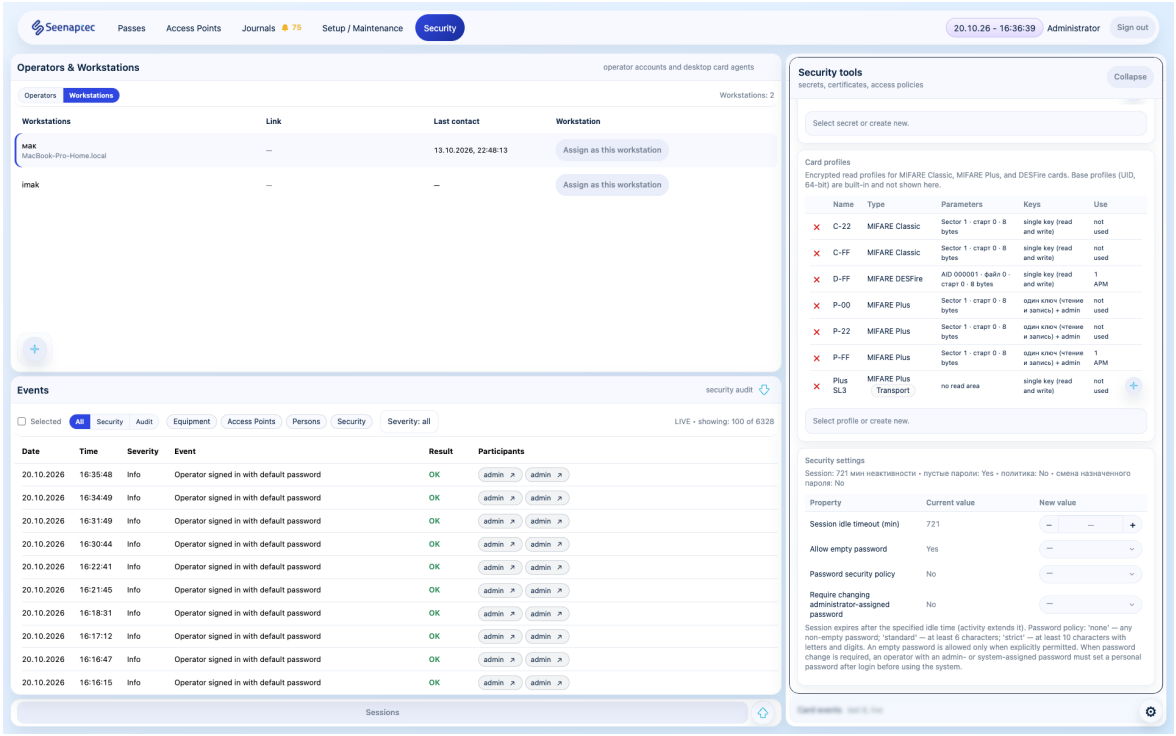
- **Download agent** — build and download the agent installer bundle (generates a new activation code);
- **Reissue distribution** — generate a new activation code: previously downloaded but unactivated bundles are invalidated, while an already activated and connected agent continues operating uninterrupted;
- **Revoke access** — immediately disconnects the agent and revokes its authorization to communicate with the server.

The **Card read profiles** section configures the ordered chain of read slots: a sequence of secure card profiles from the library (Section 14.9) that the desktop reader agent tries from top to bottom (with **Write allowed** checkboxes enabling profiles for secure card issuance and initialization dialogs), plus standard UID reading (**MIFARE (UID)**). Below the **MIFARE (UID)** row is the **Programming cards** toggle (`issuance allowed` / `issuance forbidden`) — authorizing this workstation to write reader programming and entry cards (Section 10.8); toggling takes effect immediately and requires `full` permissions on both **Reader workstations** and **Reader programming cards**, and when set to `issuance forbidden`, the server refuses to queue programming card write jobs for this workstation. The **Versions** row shows the profile set version currently applied by the agent relative to the server's latest version (`applied vX of vY`). Cryptographic keys for protected card areas are delivered to the agent from the secrets vault and reside strictly in volatile memory (RAM): read keys are held only while the Reader Workstation is bound to an active browser session, and write keys are held only while the Reader Workstation card or a card issuance/initialization dialog is open.

The **Card events** section displays a live feed of the last six card reads on this desktop reader: timestamp, extracted code, bound person, card UID, and card technology.

14.9. Card Profiles

The **Card profiles** section of the tool panel manages the library of encrypted card profiles—definitions of protected card memory areas used by SH-O field readers (Section 10.4) and Reader Workstations (Section 14.8) to read and write secure identifier codes. Base UID-reading profiles are built in and are not listed here. Working with this section requires the **Security: Card profiles** (`security.card_profiles`) permission area.



A card profile defines:

- **name** and **card type** — **MIFARE Classic**, **MIFARE Plus**, or **MIFARE DESFire**. The card type determines the cryptographic key material: MIFARE Classic uses 6-byte keys (Crypto-1), whereas MIFARE Plus and DESFire use 16-byte keys (AES-128); for MIFARE Plus cards, the target security level (**SL1** or **SL3**) is selected during Card Initialization;
- **profile kind** — **Target — keys and read area** (a full operational profile used for reading and writing cards) or **Transport — keys only** (without a read area). A transport profile serves as the source profile during card reissue and initialization for blank cards carrying factory or transport keys (Section 12.6); it is hidden from target profile selectors and reader slots, and appears first (badged **Transport**) in source profile selectors. There are no transport profiles for DESFire: factory DESFire cards are initialized without a source profile;
- **read area** (target profiles only): for MIFARE Classic/Plus — the block number (`0–1024`); for MIFARE DESFire — the 6-hex-character **Application AID** and **File ID** (`0–31`);
- **code window** (target profiles only): **Start byte** and **Code length, bytes** (`1–8`) — specifying where the card code resides within the block or file. The window must fit within a 16-byte MIFARE block, or within the first 8 bytes of a DESFire file (an SH-O reader hardware constraint);
- **keys**: **Read key** and **Write key (for issuance)** — references to secrets in the vault with the `Card key` purpose (Section 14.5); for MIFARE, you specify the key slot (**Key A** / **Key B**), and for DESFire, the application key number (`0–13`). Key material is never displayed here: you select an existing secret from the vault or create a new one inline using the add button next to the field—including server-side random key generation (**Generate**), where the raw key bytes never pass through the browser at all;
- **proximity check** — relay-attack protection for cards that support it, configurable to run **Before authentication** or **After authentication**.

For a **MIFARE Plus** profile, an additional **Plus admin keys (card level)** block is available—holding references to `CardMasterKey`, `CardConfigurationKey`, and `L3SwitchKey`. These three keys are required to personalize a factory card (transitioning from **SL0** to **SL1** or **SL3**), upgrade **SL1** → **SL3**, and rotate keys in **SL3** during Card Initialization. All three keys must be specified together as 16-byte AES-128 secrets. Without Plus admin keys, a MIFARE Plus profile can read and write cards that have already been switched to **SL3**, but cannot perform card personalization or security level transitions. MIFARE Plus security level upgrades are irreversible: once in **SL3**, a card permanently loses backward compatibility with MIFARE Classic.

To create or edit a profile, click + (**Create card profile**) at the bottom of the list or click **Edit** in the selected profile's card; click **Save** to commit changes. Modifying a profile in the library does not automatically rewrite already issued physical cards: cards written under an older revision of a profile are migrated to new keys via Card Initialization.

To delete a profile, click × on its row. An unused profile is deleted immediately. If the profile is currently referenced—in the

read chains of SH-O readers or Reader Workstations, in equipment configuration templates, or in the reader programming card set—a warning dialog lists all active bindings; clicking **Unbind and delete** removes all listed bindings and deletes the profile in a single atomic operation. Offline field devices and Reader Workstations receive the updated profile set the next time settings are written or the agent reconnects; a reader programming card set whose profile was removed will require reconfiguration (Section 10.8). Deleting a profile from the system does not alter already issued physical cards.

Chapter 15. Common Scenarios

Step-by-step procedures for everyday operational tasks. Each scenario concludes with a verification step; detailed descriptions of the screens can be found in Chapters 10–14. A typical performer role (Chapter 14) is indicated for each task; any role with the required permissions is sufficient.

15.1. Issue a Visitor Pass by Template

Role: "Reception" (or higher).

1. Open the "Passes" workspace → click the "New person" button: enter the visitor's full name and, if needed, add a photo → click "Create".
2. Select the person in the list → click "Issue pass to selected person (by template)".
3. In the dialog, select an allowed "Access template" (for example, a single-day visitor template) — the validity period, access schedule, and access objects are populated from the template and cannot be edited.
4. Add the visitor's card (present it to the desktop reader or enter the code of an existing card).
5. Click "Issue by template".

Verification: the list shows a pass with the "Active" status and today's validity period; the "Events" panel contains a pass creation record.

If no suitable templates are available ("No templates allowed for restricted issuance"), contact a security officer: the "Restricted issuance" flag on a template (Chapter 12) must be enabled by an operator with full template permissions.

15.2. Pass Request and Issuance

Roles: any operator with issuance permissions can create a request; "Pass office" issues the pass.

1. Go to "Passes" → "New pass" → set the document type switch to "Request".
2. Specify the person, validity period, and a mandatory comment (who needs access and why) → click "Create request".
3. Later, when processing the request: find it using the "Pass: Request" filter and open its card.
4. In the "Pass" section, click "Issue pass"; in the "Issue pass for request # N" dialog, assign access objects, tags, and access time, add a card → click "Issue pass".

Verification: the status changes from "Request" to "Active"; the journal records the "Pass request created" and "Pass issued from request" events.

An unneeded request can be declined using the "Reject" button in its card — the request is deleted and the action is recorded in the journal.

15.3. Revoke a Pass or Block a Person

Role: "Pass office".

Temporary suspension: in the pass card, in the bottom action bar of the "Pass" section, click "Suspend" — the change takes effect immediately without saving the card (requires full pass permissions). To restore access, click the same button, now labeled "Resume".

Pass revocation: select the pass in the list → at the bottom of the "Pass" section in the card, click the red "Delete pass" button → confirm ("Pass # N will be deleted (revoked)"). Full pass permissions are required. Pass numbers are never reused, and the history remains in the journal.

Blocking a person entirely: in the person card ("Information" section), set the state to "Blacklisted" → click "Save". No new passes can be issued to this person, and all existing passes immediately stop granting access.

Verification: an attempt to pass using a revoked or suspended pass results in "Access denied"; the "Events" panel contains change records with a before/after diff.

15.4. Check Who Opened a Door

Role: "Auditor" or "Security officer".

1. Open the "Journals" workspace and select the "Access" category.
2. Specify the time range in the "Date: from — to" fields.
3. Locate the access object: type its name in "Search", or open any event for that object and click the object chip to enable object history.
4. The feed displays all access decisions for that object; open a record to see the person, pass, identifier, and decision reason in the event card.

Verification: the "Object history" bar with the object chip is active; the records show the participants and the "Granted" result.

When investigating on the spot, the same check is even faster from "Objects" ("Access objects" panel) → select the object → "Events" panel → "Selected" toggle.

15.5. Acknowledge an Alarm

Role: "Security officer".

1. Click the bell icon on the "Journals" workspace button in the top bar (visible whenever there are unacknowledged alarms).
2. In the "Alarms" panel, select an alarm — the acknowledgment card opens on the right; review the event details and participants.
3. Enter a mandatory comment describing what was found and what action was taken.
4. Click "Confirm". For a series of identical alarms involving the same participant, use "Confirm for all (N)".

Verification: the alarm disappears from the panel and the bell counter decreases; the journal records the "Alarm acknowledged by operator" event with your comment.

15.6. Connecting a UniPOS Fire Panel and Importing Configuration

Role: "Commissioning engineer" or "Administrator".

1. Verify that the FAS subsystem is enabled ("System Overview" → tool panel → "Subsystems" → "Fire Alarm System (FAS)").
2. Open "Setup / Maintenance" and switch the **`ACS | FAS`** pill in the top bar to **`FAS`**.
3. Click **"+"** ("Add FAS panel") at the bottom of the left "Equipment" panel, enter the panel display name, IPv4 address, TCP port (`502` by default), `Modbus Unit ID` (`255` by default), and cluster panel number (`1`–`8`), then save the card (Section 10.9).
4. Wait for the panel to transition to **"Online"**: the "Communication channel" section will display the panel model (`IFS7000`), serial number, and firmware version read via Modbus `0x2B / 0x0E`.
5. In the "Service operations" section of the panel card, click **"Scan composition"** — the system automatically populates the equipment tree with system blocks (`PSU`, `BIO`, relay/input blocks), ring signaling loops, addressable devices, and fire zones (`D`, `A`, `P`).
6. In the same section, click **"Upload XML configuration"** and select the project `.xml` file exported from the UniPOS configuration utility — the system imports custom names for zones, loops, and detectors and binds detectors and modules to their respective zones.

Verification: the "Equipment" tree (`FAS` mode) displays blocks, loops, and detectors with their custom names in the "Normal" state; the "Objects" workspace (`FAS` mode) lists detection (`D`), notification (`A`), and automation (`P`) zones populated with tables of bound devices (Section 11.12).

15.7. Reset Anti-Passback Presence or a Cabinet Session

Role: "Security officer".

1. Go to "Objects" → open the "Active Access" panel.

2. “Presence” tab: locate the person in the perimeter → click “Reset presence” (or “Reset all” for the entire perimeter — for example, after an evacuation).
3. “Active sessions” tab: locate the unfinished session (a cabinet that was not closed properly) → click “Reset session”.

Verification: the entry disappears from the tab; the journal records a reset event indicating the initiating operator. The person can now enter the perimeter again.

15.8. Replace a Faulty Device

Role: “Commissioning engineer”.

Bindings to access objects reference the device record in the configuration, so you do not need to delete the faulty device record (Section 10.4).

1. Mount the replacement device on the same line with the same OSDP address. If the new device responds at a different address, update the address in the existing device record: card → “Connection” section → new address → “Save”.
2. Wait for the server to detect the hardware revision: the device transitions to the “Modified” state, and the “Model” section displays the detected differences.
3. Accept the revision (“Accept changes”). If the replacement device is of the same model, all bindings to access objects are preserved and the replacement is complete. If the model differs and the changes break existing bindings, confirm with “Accept and break bindings”, then restore the bindings in the access object card (using a binding template or manually).
4. If needed, apply a device settings template and verify the Secure Channel status (the new device may require key installation).

Verification: the device is in the “Configured” state with “Active” communication, and the access object is “Configured”; a test pass yields “Access granted”. Signals from the new reader are visible in “Commissioning (LIVE)” mode.

15.9. Update Device Firmware

Role: “Commissioning engineer”.

1. Go to “Setup / Maintenance” → tool panel → “Device firmware”: upload the firmware file into the catalog (“Upload new” or drag and drop).
2. Open the device card → “Firmware update” section: compare the current version with the available one.
3. Click “Write to device”. Monitor the progress stages: “Preparing” → “Downloading image” → “Verifying checksum” → “Transferring to device” → “Activating” → “Verifying version after write” → “Completed”.
4. For a fleet of identical devices, use the “Firmware” bulk operation (in device selection mode).

Verification: the device passport (“Model” → “Detailed”) shows the new firmware version; the journal records a “Firmware write status” event with a successful result.

15.10. Create a Backup and Restore

Role: “Administrator”.

Creating a backup:

1. Go to “Setup / Maintenance” → tool panel → “Backup / restore”.
2. Specify the backup password (a secret from the store) or enable “Unencrypted” if an unencrypted backup is required; verify the target directory.
3. Click “Create backup now”.

Verification: the journal records the “Backup creation requested” and “Backup created” events; the file appears in the backup directory (“Backup folder” parameter). For regular automated backups, enable “Auto-backup” and configure its schedule.

Restoring from a backup:

1. In the same section, choose a backup file (“Select backup”). For an encrypted backup, provide the password in the “Backup password” block (if the password is already configured in settings, the field is populated automatically).
2. Click “Recover” and confirm. Platform services restart; the panel displays their status until the system is fully ready.

Verification: after the restart, system data matches the state at the time of the backup; the journal contains events marking the start and completion of the restore operation.

Only a backup from the same installation can be restored; when migrating to a new server, restore the backup onto a clean system and complete the process by re-entering the activation key (Section 17.6).

15.11. Filter Events for a Time Period

Role: "Auditor".

1. In "Journals", set the time range in "Date: from — to".
2. Narrow the selection using a category ("Access" / "Audit" / "Technical health") and the "Domain", "Result", and "Type" filters — for example, to view all access denials for the past week, select the "Access" category and the "Denied" result.
3. To filter by a specific access object, person, or operator, add an object history filter by clicking the participant chip.
4. Tag events that require follow-up (event card → "Marks") — you can then gather them into a single view using the "Mark" filter.

Verification: the "Showing: N of M" counter reflects the filtered set; selected records open with full context.

15.12. Verify Journal Integrity and Inspect an Archive

Role: "Auditor".

Live journal:

1. Go to "Journals" → tool panel → "Journal integrity".
2. Click "Check / Verify".

Verification: the result reads "Integrity verified: events #A–#B (N pcs.)". Any violation requires immediate investigation: details are recorded in the verification event and the corresponding alarm.

Archive:

1. Tool panel → "Archive": select contiguous archive files. For encrypted archives, the password is supplied automatically from the archive settings, or you can enter it in the "Archive password" block → click "Open archive".
2. The system verifies the hash chain and file boundaries; the "ARCHIVE" indicator appears at the top along with the integrity check result (verified or violated).
3. Browse the archived events using the standard filters; when finished, click "Return to live journal".

Verification: the integrity indicator reports no violations; archived events are readable and their marks are displayed as historical.

Chapter 16. Troubleshooting

This chapter is organized by symptoms. The general diagnostic principle is top-down along the operational chain: server service → web interface → line → device → access object → pass; at each level, the system provides states and events that isolate the fault.

16.1. Web Interface Does Not Open

1. On the server, check the service status:

   ```
   sudo systemctl status srv_ctrl.service
   ```

   If the service is not in the `active (running)` state, restart it (`sudo systemctl restart srv_ctrl.service`) and inspect the service logs (`journalctl -u srv_ctrl.service`).

2. Check HTTP service readiness: `curl -i http://localhost:8080/health` — a 2xx response code is expected. If there is no response, verify whether the port is in use or has been changed in `/etc/skd/seenaptec_acs.yaml`.

3. From another computer, check network connectivity to the server and firewall rules; use the server's IP address or DNS name in the URL rather than `localhost`.

4. Verify that the MQTT broker is reachable from the server. Without the broker, the page may load, but live states, events, and the server clock in the top bar will not update.

16.2. Interface Opens, but Data Does Not Update

Symptom: lists load, but states do not change, new events do not appear, and the server date and time in the top bar are frozen.

Cause — the live data stream is interrupted: either the MQTT broker is unavailable or the browser's operational event stream connection to the server has dropped.

1. Refresh the browser page.
2. Check the MQTT broker status and its reachability from the server.
3. Check the status of platform services: "Setup / Maintenance" → tool panel → "Service Management" (or query `GET /health` to verify HTTP service readiness).

16.3. Line in "Offline", "Possible break", or "Inactive" State

- **"Inactive"** — the line is not engaged: enable "Engage" in the line card.
- **"Port unavailable"** — the service cannot open the port: for USB adapters, verify that the converter is plugged in and the port name is correct; for TCP adapters, verify that `host:port` is reachable from the server.
- **"Offline"** — the port is open, but devices do not respond: check device power, RS-485 bus polarity and continuity, and verify that the line baud rate matches the device baud rate.
- **"Possible break"** — the adapter is operating and polling is active, but no device on the line has responded for more than a minute: inspect the RS-485 bus continuity (open circuit or short circuit) and the power supply of the devices on the line.
- Inspect communication logs in the "Debug" panel (runtime log of the line's `srv_osdp` instance) and review state transition history in the "Events" panel.

16.4. Device Does Not Respond or Is Incomplete After Discovery

- The device is in the "New" state — it is not polled for operational service until accepted: open its card and click "Acknowledge" ("Accept").
- The device is disabled ("(off)") — enable "Engage".
- SSB discovery did not find all units — run discovery again: the first pass may be incomplete if some devices powered up or joined the bus with a delay.
- **Address conflict**: two devices sharing the same OSDP address on a single line cause unstable communication for both. Assign distinct addresses — for AGRG devices, addresses are assigned automatically during SSB discovery; for third-party devices, change the address in the card ("Connection" → "Change") or directly on the hardware unit.
- If a device stopped responding after changing the encryption mode, check the Secure Channel status in the "Connection" section; if the operational key was lost, reset the device key and install it again.

16.5. Access Is Denied Even Though the Pass Is Active

The exact denial reason is always recorded in the "Access denied" event — start there: "Objects" → select the access object → "Events" panel (or "Journals", "Access" category). Common reasons include:

Reason in event	What to check
"Access denied: identifier not found"	The card has not been registered or was revoked; check the code in the "Identifiers" panel
"Access denied: no permission for this access point"	The access object (or its tag) is not included in the pass
"Access denied: outside allowed schedule"	The time profile of the access object and the "Access time" of the pass
"Access denied: non-working day in production calendar"	The day type in the calendar and the "Use production calendar" setting
"Invalid PIN code"	The object's identification mode ("Card + PIN") and the person's PIN
"Access denied: escort confirmation required"	Escort confirmation mode: a person with "Requires escort confirmation" is awaiting a confirming person's card, or presented a card at an open door not covered by the current session's confirming person (Section 4.3)

Reason in event	What to check
“Access denied: escort confirmation timed out”	The confirmation window is 15 seconds; present the card again and confirm promptly
“Access denied: confirming person lacks escort privilege”	The “Can confirm escort” flag was revoked from the confirming person, or another person’s card was presented
“Access denied: confirming person has no access to point”	The confirming person does not have their own valid pass for this access object
“Access denied: escort confirmation superseded by new request”	Confirmation for another person was requested at the same point; present the card again
“Anti-passback: …”	The person’s presence state in the APB perimeter (“Active Access” → “Presence”; reset if necessary)
“Access denied: blocking condition active” / “required condition not met”	The person’s conditions and the access object’s rules
“Access denied: access point blocked”	An operator switched the object into the “Blocked” mode — reset the mode in the object card

Also check the person’s state (“Suspended” and “Blacklisted” block access), the pass state (“Suspended”, validity dates, remaining usage quota), and the identifier state (“Lost”, “Expired”).

16.6. No Access Event Appears at All

If presenting a card produces not even a denial event:

1. Check line and device communication (Sections 16.3–16.4).
2. Enable “Commissioning (LIVE)” mode in the “Events” panel of “Setup / Maintenance” and present the card: if no read signal appears, the issue is with the reader or the bus; if a signal appears, proceed to the next step.
3. Check the reader binding in the access object card (the object state must be “Configured”).
4. Verify that the access decision service is running (“Service Management” → “Access server”).

16.7. Redundancy Ring Is Not Normal

The overall ring state is indicated by the ring mode badge in the line row and line card (“Ring” row) and by the “Primary”/“Secondary” port indicators:

- **“Split: segmented polling”** — the bus is broken: devices in each segment are polled through whichever port can reach them. Inspect the cable run between the segments; once the break is repaired, devices return to the primary port automatically;
- **“Secondary adapter active”** — the primary path is completely unavailable (converter failure or lost TCP connection): check the converter and TCP reachability of the primary port; once restored, polling switches back to the primary port automatically;
- **“Primary adapter active”** — the secondary path is unavailable: check the converter and TCP reachability of the secondary port; device polling continues through the primary port, and the ring returns to normal automatically once restored;
- **“Line down”** — both paths are unavailable: check both converters and cable runs.

During a ring split or single-adapter failure, the communication state of the line and its devices remains “Active” because polling continues; the failed port is indicated by unlit status dots on its indicator (connection, RX, echo). The line displays “Port unavailable” only when both ports are unreachable.

If the ring mode is normal but some devices are still polled through the secondary port (a “Secondary” chip next to the device), this is the aftermath of a recent split: the route returns to the primary port after a hold-off delay that prevents rapid flapping.

State transition history is available in the “Events” panel (“Line ring broken: polling by segments”, “Line ring restored”, “Line adapter lost”, “Line adapter restored”, “Device poll route changed”); low-level communication details are available in the “Debug” panel (a unified log covers both ports of the ring).

16.8. “Journal integrity violation” Alarm

This alarm means that an integrity check detected a discrepancy in the journal hash chain: modification, insertion, or deletion of records, or tampering with the tail of the journal. Recording of new events continues uninterrupted.

1. Open the verification event — it specifies the violation type and the first affected record.
2. Document the circumstances (server access, timestamps) — an integrity violation indicates either direct interference with data files or storage corruption.
3. Cross-check backups and journal archives (their integrity is verified independently when opened).
4. Acknowledge the alarm with a comment summarizing the investigation findings.

16.9. Forgotten Administrator Password

If another operator with full permissions for the “Security” domain is available, change the administrator’s password from their operator card (administrator accounts can only be modified by an administrator). If administrative access is completely lost and access recovery was enabled beforehand, sign in from the login screen using a one-time vendor recovery code (Section 17.8). If access recovery was not enabled, self-service password reset is impossible — contact vendor support.

16.10. FAS Panel and Its Devices in “OFFLINE” State

When a UniPOS fire panel stops responding over Modbus TCP, the platform transitions both the panel itself and all of its child system blocks (`PSU`, `BIO`, relay/input blocks), signaling loops, addressable detectors/modules, and fire zones (`D`, `A`, `P`) to the **“OFFLINE”** state so that stale normal states are never displayed during a communication outage (Section 10.9).

Troubleshooting steps:

1. **Check the panel polling service.** Open “Setup / Maintenance” → tool panel → “Service Management” and verify that the **“FAS panel N server”** (`srv_ops`) instance is running and that “Polling enabled” is turned on in the “Connection” section of the panel card.
2. **Verify network parameters and single-client TCP limits.** In the panel card (“Connection” section), check the IPv4 address, TCP port (`502` by default), `Modbus Unit ID` (`255` for UniPOS IFS7000), and cluster panel number (`1`–`8`). Note that the Ethernet module of a UniPOS panel typically accepts only one concurrent Modbus TCP connection: if an external SCADA client or configuration utility is connected to the panel at the same time, new connections may be refused or dropped.
3. **Inspect live Modbus frames in the “Debug” panel.** Select the FAS panel in the “Equipment” tree (`FAS` mode), open the lower **“Debug”** panel, enable the **“MQTT”** toggle, and select the `DEBUG` log level:
   - **`ERR`** entries during connection or read operations (`connection refused`, `i/o timeout`, `EOF`) indicate network unreachability of the IP address, a blocked port `502`, or another client holding the TCP session;
   - outgoing **`TX fn=0x2B/0x0E`** or **`TX fn=0x03`** frames without matching **`RX`** replies indicate that the panel is not responding at the configured `Unit ID` or is timing out (if the panel is heavily loaded, increase “Response timeout, ms”);
   - **`RX EXC`** exception replies (such as `Illegal Data Address`) during polling indicate a mismatch in the configured cluster panel number or register map for the panel model.
4. Once network connectivity is restored, `srv_ops` reconnects to the panel automatically, refreshes all register states, and returns the panel, loops, devices, and zones to their live operational states.

Chapter 17. License: Activation and Lifecycle

This chapter describes the complete license lifecycle: obtaining an activation key from the vendor, activating a clean system, renewing an expired license, updating software, migrating to another server, and restoring from a backup. Dedicated sections cover the support code — the installation fingerprint that binds the license to site hardware (Section 17.7) — and access recovery using a one-time vendor code if the administrator password is lost (Section 17.8). Initial activation during installation is covered in Chapter 6; here it is presented in the context of the full lifecycle.

17.1. How Licensing Works

In Seenaptec Access Platform (SAP), the license governs **only the validity period**: functionality is always complete, with no separately licensed modules and no limits on the number of devices or workstations. License verification is performed entirely

offline — the system requires no internet connection either during activation or during regular operation.

The core principle is **soft enforcement**: an access control system must never halt physical access because of licensing. Whatever happens to the license — whether it expires or a new key has not yet been entered after an upgrade — doors and controllers continue to operate, access decisions are made, and events are recorded in the journal. The license restricts only the administrative plane: without activation you cannot sign in to the web interface to configure the system, and when the license expires the interface persistently reminds operators to renew it.

17.2. Activation Key and Two Identifiers

An **activation key** is a string of six four-character groups of hexadecimal digits (`0–9`, `A–F`):

```
VVMM-SSSS-SSSS-HHHH-HHHH-HHHH
```

The key is issued by the system vendor; it encodes the license expiration month (valid through the last day of the specified month inclusive), the installation identifier, and a cryptographic signature. You can enter the key with or without hyphens, in lowercase or uppercase — the input form normalizes it automatically. The internal structure of the key is irrelevant for day-to-day operation; what matters is that a key cannot be forged or altered — any modified key will be rejected by the system.

Every installation has **two distinct identifiers** that should not be confused:

Identifier	Origin	Purpose
Server ID (license identifier)	Extracted from the activation key; identical across all keys for this site	Links the installation to the vendor contract: the vendor issues renewal keys for this ID, and the system uses it to distinguish its own keys and backups from those of other installations
Server serial number	Generated automatically by the system on the very first service start	Internal addressing: inter-service communication, journal hash chain, and archive file names. Not used in manual configuration

Practical rule: when communicating with the vendor, always refer to the **Server ID** (it is tied to your contract); the **server serial number** is an internal technical identifier that appears in journal archive file names and diagnostics but requires no administrative action.

In addition to these two identifiers, the installation has a **support code** (installation code) — a hardware fingerprint derived from device serial numbers: it confirms the binding of the license to the physical site equipment and updates when devices are replaced (Section 17.7).

17.3. Activating a Clean System

1. Obtain an activation key from the vendor — it is supplied with the delivery or upon request under your contract.
2. Install the server package (Chapter 6) and open the web interface. On an unactivated system, the “System activation” screen is displayed.
3. Enter the key and click “Activate”.

When a valid key is entered, the system binds to the license, automatically creates the default `admin` operator account with the password `123456`, and displays the sign-in screen with the prompt: “System activated. Default login: admin / 123456 — change the insecure password after signing in.” First sign-in and mandatory password change are described in Section 6.7; alarm events generated when signing in with the default password are covered in Chapter 14.

Before activation, all platform services run normally — activation does not “turn on” hardware polling, it unlocks administrative access. However, configuring a site without activation is impossible because you cannot sign in to the interface.

17.4. Expired License and Renewal

When the license validity period expires, the system **continues to operate in full**: passages are processed, events are recorded, and operators can work in the interface. Only the interface behavior changes:

- a persistent banner appears at the top of the application: “License expired. Valid until ‹date›. The system continues to operate; contact your vendor for a new activation key”;

- a modal dialog titled “License expired” with an “Activation key” field appears periodically. Clicking “Cancel” closes the dialog and snoozes the reminder **for 5 minutes**, after which it reappears; this does not block urgent actions, but prevents the expiration from being ignored indefinitely;
- the same warning is shown on the sign-in screen.

To renew the license, request a new key from the vendor (issued for the same Server ID with a later expiration month) and enter it in the dialog — or at the next reminder prompt. Once the key is accepted, the banner and dialog disappear immediately without restarting the system.

If the entered key is cryptographically valid but its expiration date is also in the past, the system accepts it and displays: “Key accepted, but its validity period has already expired. Contact your vendor for a current key.”

17.5. Software Update and the 14-Day Grace Period

The vendor may rotate the **activation key generation** — typically alongside major product releases. If an installed update introduces a new key generation, the existing key does not become invalid immediately: a **14-day grace period** takes effect. During this period, a warning banner is displayed in the interface: “Update changed key generation. The old key remains valid for N more days — obtain a new key from your vendor.”

Administrator steps:

1. Upon seeing the banner, request a new-generation key from the vendor (the Server ID and license expiration date remain unchanged — only the key generation changes).
2. Enter the new key at any time before the grace period expires; the banner disappears immediately.

If no new key is entered within 14 days, the old key ceases to be accepted, and at the next sign-in the system prompts for a new-generation key on the activation screen. As with any other licensing condition, expiration of the grace period has no impact on doors or controllers.

Both the start and the expiration of the grace period are recorded in the event journal; expiration raises an alarm event.

17.6. Migration, Restore, and Reactivation

The activation key is stored on the server in protected local storage and is **excluded from standard backups** — you can store and transfer backup files without risking key exposure. This design defines the rules for migration and restore. A separate case is a backup created with the “Include secrets” option enabled: it contains all system secrets, including the activation key, and is always encrypted (Section 10.8); treat such a backup with the same care as a password.

Restoring on the same installation (same server, standard rollback to a backup) requires no licensing steps: the key remains in place and the Server ID matches.

Migrating to a new server: install the server package on the new machine and restore the backup **onto a clean, not-yet-activated system** — this is permitted. Because the new machine has no key in local storage, the system enters reactivation mode after the restore completes: the activation screen displays the message “Activation key lost. Enter the key for this installation (same server ID) — operator accounts are preserved.” Enter the same key that was used previously (or request it again from the vendor). The default `admin` operator is **not created** during reactivation — all operator accounts come from the restored backup, and operators sign in with their existing passwords. If the backup was created with “Include secrets” enabled, the activation key is restored automatically along with the other secrets, and reactivation is not required.

Foreign backup — a backup taken from a different installation (with a different Server ID) — **cannot be restored**: the system rejects the restore operation. This prevents accidental mixing of data from different sites; merging or copying one installation into another via backup is not supported.

A key issued for a different Server ID is also rejected during activation: the form reports “Key was issued for another installation (different server ID).”

17.7. Support Code (Installation Fingerprint)

The **support code** (labeled “Installation code” in the UI) is a short installation fingerprint that binds the license to the physical hardware on site. The code is not secret and grants no access to the system: it serves as an identifier for this specific installation. Vendor support compares the code you provide against the customer record to verify that a support request pertains to the contracted installation.

The fingerprint is computed from a core set of devices — up to five devices that are accepted into the configuration, engaged, bound to access objects, and actively communicating. The system selects and maintains this core set automatically; it cannot be edited manually. The code consists of four-character hexadecimal groups separated by hyphens — one group per core device, for example `F5A2-CECD-2026-F0B2`.

Where to Find the Code

1. Click the system logo in the top-left corner of the screen to open "System Overview".
2. Open the tool panel using the gear button near the bottom-right edge of the screen.
3. Locate the "Support" section.

The section displays three values:

- **Server ID** — the license identifier from the activation key (Section 17.2); if the system is not activated, a dash is shown instead;
- **Installation code** — the hardware fingerprint itself;
- **Serial number** — the server serial number (Section 17.2).

The "Copy all" button copies all three values as a single text block that can be pasted directly into an email or support ticket. When contacting support, provide the **Server ID and Installation code**: support uses the Server ID to locate your installation record and the Installation code to verify the hardware binding.

The collapsible "Fingerprint composition" block lists the core devices: code group, line and device address, and device serial number. The note "offline for a long time" indicates that a device has not communicated for more than two weeks; if it remains offline for a month, the device drops out of the fingerprint and the code changes.

When the Code Changes

The code changes whenever the core device set changes: when a faulty device is replaced with a new unit (Section 10.4) or when a device remains offline for more than a month and drops out of the fingerprint. The date of the last change is shown in "Fingerprint composition" as "active since …". After replacing a device that was part of the fingerprint, report the updated code to support so that future verification checks match.

The message "Device status is updating after restart." appears immediately after a server restart while the system has not yet polled all devices and is displaying the code based on the saved composition. This is normal: the displayed code is valid, and the notice disappears after the first polling cycle.

Why the Code May Be Absent

The fingerprint is generated only when at least two devices are commissioned and operating. While fewer than two devices are active, the message "Code will appear after commissioning at least 2 devices" is shown instead — the same message appears on a fresh system before equipment is polled for the first time. Accept devices into the configuration, engage them, and bind them to access objects (Chapter 10) — the code will appear automatically.

17.8. Access Recovery Using a Vendor Code

Access recovery provides an emergency sign-in path when the `admin` password is lost or the `admin` account has been deleted and no other operators with full permissions remain. Sign-in is performed using a one-time code issued by the vendor after verifying the customer's identity under vendor security procedures. The server stores only the cryptographic hash of the expected code: neither reading the database nor inspecting a backup reveals the recovery code, which the vendor computes from a secret key not included in the product.

Security policy: keep access recovery disabled during normal operation and enable it only for the duration of a support session. While access recovery is disabled, signing in with a recovery code is impossible in principle. Enable it in advance when engaging vendor support for maintenance work: enabling recovery requires active administrative access, so it cannot be turned on after the password is already lost. Disable access recovery as soon as the maintenance work is complete.

Enabling and Disabling

1. Request the **first code hash** from the vendor — support computes it for your Server ID (Section 17.2) and records it in your installation profile.

2. Open "System Overview" (click the system logo) → tool panel → "Access recovery" section (located between "Support" and "Attention thresholds").
3. Paste the received hash into the "First code hash" field — a 64-character hexadecimal string (SHA-256); letter case and extra separators are ignored — and click "Enable".

The section displays the current status — "Enabled" (with the hash update timestamp) or "Disabled" — and, when enabled, the **expected code hash** itself: compare it with the vendor's record to confirm that both sides are synchronized. The same input field is used to rotate the hash ("Replace hash" button) — for example, if the code chain goes out of sync. Clicking "Disable" prompts for confirmation and turns off access recovery: the stored hash is deleted, and re-enabling recovery later will require requesting a new hash from the vendor.

Viewing this section requires the "Read" level in the "Operators" area of the "Security" domain; enabling, replacing the hash, and disabling require the "Full" level (Chapter 14).

Signing In with a Recovery Code

1. Contact the vendor. After verifying your request, support provides a **recovery string** formatted as `CODE.HASH` — the one-time recovery code and the hash of the next code combined into a single string.
2. On the sign-in screen, click "Recover access" — this button is visible only when access recovery is enabled.
3. Enter the installation's **activation key** and the **recovery string**, then click "Recover". Both values are required: because the activation key and the recovery string travel through separate channels (contract documentation vs. support communication), intercepting a single message does not grant access.
4. When both values are valid, the system resets the `admin` password to the default value (re-creating the `admin` account if it was deleted), terminates all existing `admin` sessions, and immediately opens a new session with a mandatory password change prompt — set a new password just as on first sign-in (Section 6.7).

Each recovery code is strictly single-use: upon successful sign-in, the system stores the next-code hash extracted from the recovery string, so the same string cannot be used a second time. If another recovery is needed later, the vendor issues the next string in the chain — the code chain advances forward, rendering previously used messages useless.

If the activation key does not match, the form reports "Invalid activation key. Check your input."; if the recovery string is invalid, it reports "Invalid recovery code. Check the string provided by the vendor." If an administrator disables access recovery between page load and submission, the form reports: "Access recovery is disabled by administrator."

Journal Audit and Brute-Force Protection

Every recovery action is recorded as an alarm event (Section 13.4): "Access recovery enabled", "Access recovery disabled", "Access recovery completed", "Failed access recovery attempt", and "Access recovery locked by brute-force protection". After several failed attempts, the system enforces an increasing cooldown delay between attempts — the form displays "Too many recovery attempts. Retry in …" along with the remaining wait time; the failure counter is persisted on the server and is not reset by a service restart.

17.9. Common Situations

We lost our activation key. An activation key is not a one-time secret — the vendor retains records of issued keys and will re-send your installation's key upon request under your contract. While the system is running, losing the written copy of the key has no effect because the key is stored on the server. You will only need the key again when migrating to a new server or restoring onto a clean system (Section 17.6).

We forgot the `admin` password and there are no other administrators. If access recovery was enabled, request a recovery string from the vendor and sign in using the "Recover access" button on the login screen (Section 17.8). If access recovery was not enabled, self-service recovery is not possible — contact the vendor.

We are migrating the server to new hardware. Create a fresh backup, install the package on the new machine, restore the backup onto the clean system, and re-enter the installation's activation key (Section 17.6). During the migration window, doors operate within the standalone capabilities of the field hardware; once the new server starts, normal operation resumes with the data from the backup.

The license expired over the weekend. Nothing urgent happens: doors continue to operate, events are logged, and alarms can be acknowledged. Operators will see the expiration banner and a periodic renewal dialog — clicking "Cancel" snoozes it for 5 minutes. On the next business day, request and enter a renewal key (Section 17.4).

We updated the software and a yellow banner about key generation appeared. This is the 14-day grace period (Section 17.5): you have 14 days to obtain a new-generation key from the vendor and enter it. Your license expiration date and Server ID remain unchanged.

The system rejects a key with a message about another installation. The key you entered belongs to a different site or contract. Verify with the vendor which Server ID is assigned to this installation and obtain the matching key.

Can the system be operated without activation? Platform services run, but you cannot sign in to the web interface or configure the system — on a clean installation there are neither operators nor configuration data. Activation is the mandatory first step of commissioning (Chapter 6).

Support asks for the "support code" — where do I find it? Open "System Overview" (click the system logo) → tool panel → "Support" section, and read out the Server ID and Installation code (Section 17.7). If you recently replaced any devices, the code may have changed — always provide the current value shown in that section.

Chapter 18. API Integration

All system functions are accessible via the REST API — the exact same interface used by the web application. This chapter describes the general integration workflow: where to find the specifications, how an external system authenticates, and which permissions are required for common integration tasks. Individual API endpoints are not listed in this guide — they are documented in full in the OpenAPI specifications.

18.1. Specifications and Swagger UI

Resource	URL
Swagger UI — interactive browsing and method invocation	`http://<server-address>:8080/api-docs/`
OpenAPI specification files and documentation	`http://<server-address>:8080/docs/`

The API is organized into functional areas, each served under its own path prefix:

Area	Prefix	Contents
Equipment	`/api/hw/`	Server, lines, devices, access objects, bindings, service status
Access data	`/api/perm/`	Persons, identifiers, passes, templates, tags
Journal	`/api/journal/`	Events, alarms, marks, operational event stream
Authentication and security	`/api/auth/`	Sign-in, operators, roles, sessions, security settings

Server readiness can be checked with `GET /health` (returns `ok`; no authentication required).

18.2. External System Authentication

An external system authenticates in the same way as a human operator: a dedicated operator account is created for the integration and assigned a role scoped to the required permissions (Chapter 14). Workflow:

1. An administrator creates a role with the minimum required permission set and a dedicated operator account for the external system.
2. The external system signs in (`POST /api/auth/api/v1/login` with login and password) and receives the `skd_operator_session` session cookie.
3. All subsequent requests include this cookie. The session is extended on each request; if idle longer than the configured inactivity timeout (set in "Security settings", Chapter 14), the session expires and the client must sign in again.

All actions performed by an integration account are recorded in the audit journal alongside human operator actions, and its active sessions are visible in the "Sessions" panel of the "Security" workspace.

18.3. Permission Model

Permissions are addressed by a three-part key: `domain → area → level`. There are five domains:

Domain	Scope
`equipment`	Equipment: server, lines, devices, system maintenance operations
`access`	Access objects, rules, active access
`iam`	Persons, identifiers, passes, templates
`security`	Operators, roles, secrets, reader workstations, card profiles, platform audit
`journal`	Event journal, alarms, integrity verification, archive

Permission levels are cumulative — each higher level includes all preceding ones:

- `read` — read access;
- `issue_tpl` — applicable only to the passes area: pass issuance strictly from templates marked for restricted issuance;
- `add` — record creation;
- `full` — modification and deletion.

General mapping to HTTP methods: `GET` requires `read`, `POST` requires `add`, and `PUT`/`DELETE` require `full`. Exact permission requirements for each endpoint are specified in the OpenAPI documentation.

18.4. Permissions for Common Integration Tasks

Task	Sufficient permissions
Read the event journal (full journal)	`journal.events.read`
Read events for a single domain (for example, access events only)	read permission for that domain (for example, `access.config.read`)
Equipment monitoring: line and device states	`equipment.devices.read`
HR synchronization: creating persons	`iam.persons.add`
Issue passes using allowed templates	`iam.passes.issue_tpl` + `iam.persons.read`
Full pass management (including revocation)	`iam.passes.full`
Read active access state (perimeters, sessions)	`access.config.read`

Integration design principle: grant the role with the minimum sufficient permission level. For example, an HR system that registers employees and issues standard passes only needs `iam.persons.add` and `iam.passes.issue_tpl` — it will not be able to modify equipment, system settings, or unrelated records.

18.5. Implementation Notes

- **Commands are asynchronous.** A successful HTTP response to a hardware command means the command was accepted for execution, not that the physical operation has completed; the actual outcome is confirmed via state updates or journal events (Chapter 8).
- **Record identifiers are never reused.** Numeric `id` values of persons, passes, identifiers, and other entities are assigned by the server and are never reallocated after deletion — external systems can safely store them as persistent foreign keys.
- **Core fields of an issued pass are immutable.** Extending validity, replenishing usage quota, binding identifiers, and changing status are permitted; the person, start date, and set of access objects of an issued pass cannot be modified (Chapter 12).
- **Live data** is available via the operational event stream (Server-Sent Events):

  `GET /api/journal/api/v1/journal/stream/operational`

  Permission checks for requested domains follow the same rules as historical journal queries.

Chapter 19. Company Details and Contacts

This edition of the guide is intended for international deployments and English-language operation.

Company Name	Seenaptec LLC (ООО «Синаптек»)
TIN (INN)	7717776110
KPP	771601001
OGRN	1147746137769
Address	129343, Moscow, proezd Serebryakova 8, bldg. 1, fl. 2, prem. I, office 6, Russian Federation
Phone	+7 (495) 255-27-35
E-mail	kf@seenaptec.com

Skolkovo Innovation Center

