

# T3 — Cosyma Unified Info-System (Devices • Patients • Support • MediaSync)

Версия: 0.9 (2025-08-30) Ответственный: Никита Горелов

---

## 1) Назначение и цели

**Задача:** создать единую инфо-систему (далее — *CUIS*), обеспечивающую независимый контроль над двумя главными сущностями Cosyma: **приборы** (медицинское оборудование) и **пациенты**. Система должна поддерживать расширяемый веб-интерфейс, ролевую модель доступа, интеграцию с Telegram-ботом для регистрации/саппорта приборов и связь с сервисом **MediaSync** для работы с медиа и артефактами (протоколы, отчёты, телеметрия).

**Ключевые цели:** 1. Прозрачный инвентарный учёт и жизненный цикл **приборов** (местоположение, статус, владельцы/пользователи, сервис, RMA). 2. Полная карта **жизненного цикла пациента** (фазы, события, состояния), включая тесты/оценки, протоколы терапии, сессии и отчёты. 3. Единая поддержка и регистрация через **Telegram-бот**, связанная с учётом приборов и карточками пациентов. 4. Сквозная работа с **медиа/артефактами** (MediaSync): прикрепление к пациентам, приборам, тестам, сессиям; экспорт и доступ по ролям. 5. Экспорт данных (CSV/JSON/PDF), аудит действий и готовность к дальнейшему масштабированию.

---

## 2) Область работ (Scope)

Входит в MVP (этап 1): - Бэк-офисное веб-приложение с вкладками **Devices** и **Patients**, плюс **Support** (заявки) и **Media** (просмотр связей/поиска по MediaSync). - Базовая **RBAC** (роли и права), авторизация, аудит. - **Воркфлоу** пациента: онбординг → оценка → терапия → контроль/выписка → сопровождение → повтор. - **Воркфлоу** прибора: регистрация → инвентаризация → назначение/перемещение → сервис/RMA → списание. - **Telegram-бот** для регистрации приборов, открытия тикетов саппорта и FAQ. - **Интеграция с MediaSync**: привязка медиа/файлов/телеметрии к сущностям.

Вне MVP (этап 2+): - Продвинутое отчёты, дашборды, KPI/коHORTы. - Машинные подсказки по выбору протоколов, риск-флаги. - Мультиклиентский режим (дилеры/партнёры), биллинг, договоры.

---

## 3) Роли и доступы (RBAC)

**Администратор** — полные права, управление ролями/настройками. **Врач** — доступ к пациентам, протоколам, отчётам, назначению приборов пациенту. **Методист/Ассистент** — ведение сеансов, тестов, журналов, телеметрии, медиавложений. **Техник (Service)** — учёт приборов, перемещения, сервис/RMA, смежные тикеты. **Саппорт-агент** — обработка тикетов, ответы пользователям,

эскалации. **Продажи/Операции** — просмотр инвентаря, статусы поставок/закупок, контакты. **Пациент (опц.)** — портал пациента (этап 2+): просмотр расписания/отчётов.

Доступы настраиваются по сущностям и действиям:  
read/create/update/archive/assign/export.

---

## 4) Ключевые сущности и атрибуты

### 4.1 Device (прибор)

Идентификаторы: `id`, `serial`, `qr_code`, `external_ids[]`. Справочники: `model`, `hardware_rev`, `firmware_version`. Статусы: `registered` | `in_stock` | `at_clinic` | `at_patient_home` | `under_service` | `RMA` | `decommissioned`. Привязки: `current_location` (клиника/склад/адрес), `owner` (кто закупал), `assigned_patient_id?`, `responsible_user` (техник/врач), `warranty_until`, `purchase_order`. Эксплуатация: `last_seen_at`, `last_sync_at`, `telemetry_endpoint`, `maintenance_notes[]`. Документы/Медиа: ссылки на MediaSync (паспорта, акты, фото, лог-дампы).

### 4.2 Patient (пациент)

Идентификаторы: `id`, `mrn?`, `full_name`, `dob`, `sex`, `contacts[]`, `consents[]`. Медицинские: `primary_diagnosis`, `comorbidities[]`, `contraindications[]`, `clinic`, `attending_physician`. Жизненный цикл: `phase`, `state`, `next_checkpoint_at`. Оценки/тесты: список записей `AssessmentInstance` (по шаблонам `AssessmentTemplate`). Протокол/Курс: `ProtocolAssignment` (параметры, цели), `Session[]` (с телеметрией/замечками), отчёты. Медиа/Артефакты: клинические изображения, видео ходьбы, документы согласий.

### 4.3 Assessment (оценки/тесты)

`AssessmentTemplate` (шаблон шкалы, типы вопросов: числовые/качественные/шкальные/чек-лист) и `AssessmentInstance` (результаты, привязка к пациенту + времени). Примеры: mRS, Barthel, MoCA, 5MWT, локальные тесты ЛФК, спец-тесты по нозологиям. Поля: `score`, `subscores`, `attachments` (видео), `interpretation`.

### 4.4 Protocol / Session / Telemetry

`Protocol` (библиотека, версия, параметры), `ProtocolAssignment` (назначено пациенту), `Session` (факт сеанса, параметры применены/отклонения, субъективные ощущения), `Telemetry` (сырые и агрегированные данные, источник: прибор/ручной импорт).

### 4.5 Clinic / Contact / User

`Clinic` (адреса, отделения), `Contact` (покупатель, заведующий, техник), `User` (аккаунт в системе, роль).

## 4.6 Ticket (саппорт)

**Ticket** (создан из Telegram-бота/веба), поля: **category** (регистрация/настройка/кабель/ошибка/другое), **status** (new/in\_progress/waiting/resolved/closed), **priority**, **device\_id?**, **patient\_id?**, **attachments** (фото/видео), **public\_thread** (ответы бота), **internal\_notes**.

## 4.7 Media (MediaSync)

**MediaObject** (id, type: image/video/doc, tags, sha256, storage\_uri, previews), **Linkage** (media\_id ↔ entity:id + role: evidence/report/telemetry\_dump/etc.).

---

# 5) Жизненные циклы и состояния

## 5.1 Пациент

Базируется на согласованной модели фаз/событий/состояний (см. «Patient Lifecycle»). Для MVP — кодированные **фазы** и **состояния**, со сменой через события. Примеры состояний: стационар, готовится к реабилитации, первичная диагностика, активная терапия, приостановка, выписка, сопровождение, повторный цикл, архив. Настраиваемые правила уведомлений и чек-поинтов.

## 5.2 Прибор

Фазы: поставка/приёмка → склад/инвентарь → назначен клинике → назначен пациенту (дом/центр) → сервис/RMA → возврат/склад → списание. События: **registered**, **assigned\_to\_clinic**, **assigned\_to\_patient**, **moved**, **maintenance\_started/finished**, **rma\_opened/closed**, **decommissioned**.

Каждая смена состояния фиксируется в **AuditLog** + публикуется доменное событие (см. §9).

---

# 6) Ключевые сценарии (MVP)

## S1. Учет и перемещение прибора

- Регистрация прибора (веб/бот QR): ввод серийника, модели, ревизии, привязка к локации/клинике, загрузка паспорта (MediaSync).
- Перемещение: смена локации (склад → клиника/дом), назначение ответственных, печать акта (PDF), уведомление техника.
- Сервис/RMA: открытие тикета, статусы, история, вложения, возврат.

## S2. Карточка пациента и протокол

- Онбординг: ввод данных, согласия, прикрепление первичных документов (MediaSync), назначение врача.
- Оценка: заполнение шкал (веб-формы), прикрепление видео/фото, вычисление скорингов.

- Назначение протокола: выбор из библиотеки/кастом, цели, точка отсчёта, план сессий.
- Проведение сеансов: учёт факта/параметров/самочувствия, телеметрия, промежуточная оценка, коррекция.
- Выписка и отчёты: финальные тесты, общий отчёт (PDF), рекомендации, план сопровождения.

### S3. Telegram-бот (регистрация/саппорт)

- `/start` — выбор роли (врач/техник/клиника/пациент). Авторизация по коду/номеру договора/QR.
- **Регистрация прибора:** скан QR → подтверждение → создание записи `Device` /линк к `Clinic` /контакту.
- **Саппорт-тикет:** категория → вопросы triage → сбор фото/видео → создание `Ticket` → уведомления/ответы.
- **Статусы:** запрос статуса тикета/прибора, базовые FAQ.

### S4. MediaSync

- Загрузка медиа в MediaSync (из веба/бота), присвоение тегов и привязок.
- Просмотр всех медиа в карточке пациента/прибора; предпросмотр + скачивание по ролям.

## 7) Требования к UI (веб)

### 7.1 Навигация и общие принципы

- Вкладки: **Devices**, **Patients**, **Support**, **Media**, **Admin**.
- Глобальный поиск по `serial/name/clinic/ticket` и теги.
- Списки: табличные виды с сохранением фильтров/видов, быстрые действия.

### 7.2 Devices (список)

Колонки: Serial, Model, Status, Location, Assigned (Clinic/Patient), Last Sync, Warranty, Owner, Actions. Фильтры по статусам/локациям/модели. **Карточка прибора:** паспорт, история перемещений (timeline), текущие назначения, активные тикеты, связанные медиа/телеметрия.

### 7.3 Patients (список)

Колонки: Name, Diagnosis, Phase/State, Clinic, Protocol, Next Checkpoint, Assigned Device, Last Update. **Карточка пациента:** демография, клиника/врач, **Лента** (тесты/сессии/телеметрия/медиа), текущий протокол, цели, **Жизненный цикл** (визуальный трек), кнопки быстрых действий (оценка/сеанс/отчёт).

### 7.4 Support

Список тикетов с фильтрами по статусу/приоритету/категории. Карточка с публичным тредом (бот) и внутренними заметками.

## 7.5 Media

Галерея/таблица MediaSync: превью, теги, связи; быстрый переход в карточку сущности.

## 8) Датамодель (укрупнённо)

```
User (id, role, clinic_id?)
Clinic (id, name, address, contacts[])
Device (id, serial, model, hw_rev, fw_ver, status, location_ref, owner_ref,
assigned_patient_id?, warranty_until, ...)
Patient (id, name, dob, sex, contacts[], diagnosis, comorbidities[], clinic_id,
physician_id, phase, state, next_checkpoint_at, ...)
AssessmentTemplate (id, code, name, version, schema)
AssessmentInstance (id, patient_id, template_id, occurred_at, payload, score,
attachments[])
Protocol (id, code, name, params_schema)
ProtocolAssignment (id, patient_id, protocol_id, goals[], baseline_ref, active)
Session (id, patient_id, assignment_id, device_id?, occurred_at, params, notes,
subjective, attachments[])
Telemetry (id, device_id, session_id?, captured_at, raw_uri, metrics)
Ticket (id, category, status, device_id?, patient_id?, priority, public_thread,
internal_notes)
MediaLink (id, media_id, entity_type, entity_id, role)
AuditLog (id, actor_id, entity_type, entity_id, action, before, after,
occurred_at)
EventOutbox (id, type, payload, published_at?)
```

schema / payload — JSON Schema для гибкости тестов и параметров протоколов.

## 9) API и события

### 9.1 REST (MVP)

- POST /auth/login
- GET/POST/PATCH /devices, POST /devices/{id}/assign, POST /devices/{id}/move, POST /devices/{id}/service.
- GET/POST/PATCH /patients, POST /patients/{id}/assessments, POST /patients/{id}/protocols, POST /patients/{id}/sessions.
- GET/POST /tickets, PATCH /tickets/{id}.
- GET/POST /media-links, GET /media/{id} (прокси к MediaSync или presigned-URL).
- GET /lookups (модели/шаблоны/клиники/пользователи).

## 9.2 Вебхуки / доменные события (outbox)

- patient.state.changed, patient.assessment.recorded, patient.session.recorded, patient.report.published.
- device.assigned, device.moved, device.service.started/finished, device.rma.opened/closed.
- ticket.created/updated.

Схемы сообщений — JSON Schema v2020-12. Доставка через очередь (NATS/RabbitMQ) или вебхуки партнёрам (этап 2+).

---

## 10) Telegram-бот (интеграция)

Транспорт: Bot API (вебхуки). Разделение команд по ролям и контексту чата.

**Команды (MVP):** - /start → выбор роли/языка, привязка к учётной записи/клинике. - /register\_device → скан/код → проверка серийника → создание/линк. - /support → категория → пошаговый опрос → фото/видео → тикет. - /status → по тикету/прибору.

**Безопасность:** одноразовые PIN/коды регистрации, связывание с User, ограничение админ-команд, логирование.

---

## 11) Интеграция с MediaSync

- **API MediaSync** (существующий или доопределить): POST /upload, GET /media/{id}, GET /search?q=..., POST /link {media\_id, entity}.
  - **Метаданные:** tags[], entity\_type, entity\_id, role, sha256, content\_type, created\_at.
  - **Механика превью** для видео/изображений, дедупликация по хешам, хранение в S3-совместимом объектном хранилище.
- 

## 12) НФ-требования и безопасность

- **Доступность:** 99%+ для бэк-офиса MVP.
- **Производительность:** списки до 50k записей (серверная пагинация/поиск), загрузка медиа до 2 Гб через резюмируемый аплоад.
- **Языки:** RU/EN (этап 1), TH (этап 2).
- **Логи и аудит:** действия пользователей, смены статусов, экспорт/скачивание.
- **Резервное копирование:** ежедневные снапшоты БД, версии медиа-метаданных, тест восстановления ежемесячно.
- **Конфиденциальность:** разделение PHI/PII, шифрование в покое (AES-256 в хранилище) и в транзите (TLS), полевая маскировка.

- **Соответствие:** базовые требования 152-ФЗ (РФ) и PDPA (ТН) — хранение согласий, ограничение доступа, протокол инцидентов.
- 

## 13) Технологический контур (базовый ориентир)

Указать как **рекомендуемый базовый стек**, допускается замена по инфраструктуре Cosyma. - **Backend:** TypeScript, NestJS, PostgreSQL, Prisma; очереди NATS/RabbitMQ; OpenAPI. - **Frontend:** Next.js (App Router), React-UI kit (например, shadcn/ui) + Tailwind; i18n; SSR + RSC. - **Бот:** Telegraf (Node), вебхуки. - **Хранилище медиа:** S3-совместимое (Selectel, VK Cloud, Yandex Object Storage, minio on-prem). - **Деплой:** Docker Compose (MVP) → Kubernetes (этап 2+); CI/CD (любая локально доступная: Woodpecker/GitLab CE/Gitea Actions).

Альтернативы без привязки к VPN: on-prem Linux сервер, локальные реестры Docker, внутренний Git (Gitea).

---

## 14) Экспорт/отчётность

- **Экспорт CSV/JSON** для списков Devices/Patients/Tickets с фильтрами.
  - **PDF-отчёты:** промежуточные и итоговые по пациенту (включая графики прогресса и сводку тестов), акты перемещения устройства, сервисные отчёты.
- 

## 15) Приёмка (Acceptance) — MVP чек-лист

**Функциональные:** 1. CRUD по Devices/Patients + фильтры/поиск. 2. Регистрация прибора через бота → запись в CUIS, видимость в UI. 3. Создание тикета через бота → отображение и обработка в UI. 4. Ведение тестов и сеансов по пациенту, генерация промежуточного отчёта (PDF). 5. Привязка медиа (MediaSync) к пациенту, прибору, тесту/сеансу; предпросмотр. 6. Аудит-лог ключевых действий, экспорт списков (CSV).

**Нефункциональные:** 7. RBAC и запрет действий вне прав. 8. Бэкапы БД восстанавливаются на стенде. 9. Нагрузка: выборка 10k устройств < 2 сек (кэш + индекс), загрузка медиа до 1 ГБ устойчиво.

---

## 16) План релизов

**MVP (4–6 недель ориентир):** - Нед. 1–2: ядро БД/моделей, Devices CRUD, базовый UI, аутентификация, MediaSync интеграция (минимум: линковка, просмотр). - Нед. 3–4: Patients (оценки, протокол, сессии), PDF отчёт v1, RBAC, аудит. - Нед. 5: Telegram-бот (регистрация/тикеты), Support UI, экспорт CSV. - Нед. 6: стабилизация, нагрузочные/безопасность, документация и приёмка.

**Этап 2:** дашборды, портал пациента, расширенные вебхуки/интеграции, мульти-клиент.

---

## 17) Открытые вопросы и риски

- Формализация шаблонов оценок (полные формы, шкалы, интерпретации).
  - Стандарты телеметрии и частота/формат синхронизации.
  - Политика хранения медиа (сроки, анонимизация, маскирование видео-лиц при обмене).
  - Регламенты RMA и сервисной логистики (SLA, статусы, документы).
  - Нюансы PDPA/152-ФЗ и трансграничная передача данных.
- 

## 18) Приложения (структуры JSON Schema, контуры UI)

### 18.1 Пример AssessmentTemplate.schema

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "type": "object",
  "properties": {
    "fields": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "key": {"type": "string"},
          "label": {"type": "string"},
          "type": {"enum": ["number", "string", "select", "boolean", "file"]},
          "required": {"type": "boolean"},
          "options": {"type": "array", "items": {"type": "string"}}
        },
        "required": ["key", "label", "type"]
      }
    }
  },
  "required": ["fields"]
}
```

### 18.2 Событие patient.state.changed

```
{
  "type": "patient.state.changed",
  "id": "evt_...",
  "occurred_at": "2025-08-30T09:00:00Z",
  "patient": {"id": "pat_123", "prev_state": "diagnostics", "new_state":
    "active_therapy"},
}
```



```
"reason": "assessment_completed"
}
```

### 18.3 Событие `device.assigned`

```
{
  "type": "device.assigned",
  "id": "evt_...",
  "occurred_at": "2025-08-30T09:00:00Z",
  "device": {"id": "dev_456", "serial": "S123"},
  "assigned_to": {"type": "clinic", "id": "cln_1"}
}
```

---

Готово для импорта в Cursor.