ИНТЕР РАО ЕЭС

энергия без границ

Малая автоматизация в энергетике

Платформа безопасной citizen-разработки | On-premise | Контейнерная инфраструктура

Июль 2026

ПОВЕСТКА

01 КОНТЕКСТ И ВЫЗОВЫ

- ИИ меняет подход к автоматизации
- Риски и дилемма
- Концепция решения

02 ПЛАТФОРМА АВТОМАТИЗАЦИИ

- Markdown + Mermaid.js
- Контейнерное облако
- Шаблонные проекты

03 БЕЗОПАСНОСТЬ КАК SELF-SERVICE

- GitLab-репозиторий
- SAST + SCA автоматика
- Шаблоны с Методикой ИБ

04 ДОРОЖНАЯ КАРТА 2027–2029

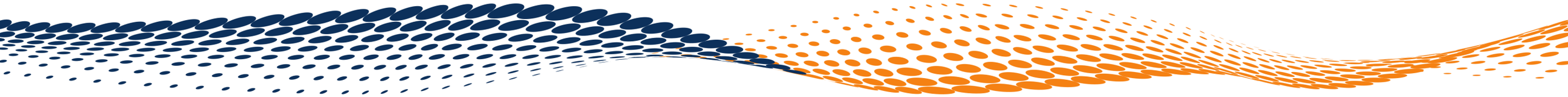
- 2027: Фундамент
- 2028: Платформа
- 2029: Масштабирование

ИНТЕР РАО ЕЭС

энергия без границ

01. КОНТЕКСТ И ВЫЗОВЫ

Почему сейчас? Как ИИ меняет подход к операционной автоматизации и какие риски это порождает



ИИ снизил порог входа до пары промптов — новая реальность

Взрывная скорость разработки

С появлением ИИ-инструментов для разработки время создания микросервисов сократилось на порядки. Перекладывание документов, их анализ, автоматизация действий в ИС — теперь доступно бизнес-пользователям.

Операционная эффективность

Операционные затраты компании могут быть резко оптимизированы за счёт быстрой citizen-разработки. Каждый сотрудник может автоматизировать свою рутину.

Вывод для ИТ и бизнеса

Блок ИТ и Компания должны **наращивать темпы малой автоматизации**, а не тормозить её. Игнорирование этой тенденции означает упущенную эффективность и теневые решения.

КЛЮЧЕВОЙ ПОКАЗАТЕЛЬ

10x
ускорение разработки с ИИ-инструментами

пары промптов
для создания рабочего микросервиса

Дилемма: операционная эффективность vs контроль и безопасность

Зоопарк скриптов

Нет стандартов и контроля качества. Каждый скрипт — уникальное решение, неподдерживаемое коллегам.

Потеря знаний

При увольнении автоматизатора уходят ценные знания. Нет компенсирующих мер — бизнес-процессы разваливаются.

Дыры в информационной безопасности

Непроверенный код получает доступ к корпоративным системам. Малая автоматизация становится угрозой вместо актива.

Нужен баланс: эффект + контроль

Без должного контроля малые автоматизации становятся теневыми решениями. Задача ИТ — не запретить, а создать **конвейер**, где автоматизация быстрая, безопасная и документированная.

Ключевой принцип: разработчик готовит только код. Всё остальное — инфраструктура, ИБ, документация — берёт на себя Блок ИТ.

Решение: конвейер документирования + ИБ-контроль, адаптированный под реальность

1 — Снижает порог входа

Markdown вместо Word, Mermaid вместо Visio, ИИ-помощник вместо техписателя

2 — Обеспечивает корпоративный контроль ИБ

SAST + SCA автоматика, hardened-контейнеры, контролируемый репозиторий

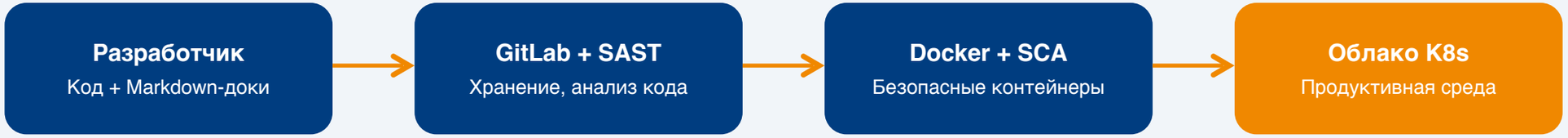
3 — Сохраняет знания

Централизованный GitLab, шаблонные проекты, независимость от кадров

4 — Адаптирует ВНД под реальность

Текущие ВНД + современные возможности ИИ и контейнеризации

Схема конвейера: разработчик → Блок ИТ → продуктив



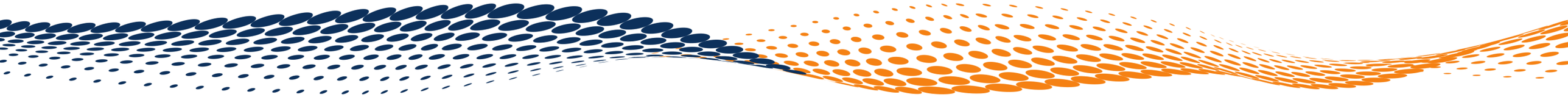
Разработчик готовит **только** код и функциональную документацию в Markdown. Всё остальное — инфраструктура, сеть, защита ОС, проверки ИБ — берёт на себя **Блок ИТ** как держатель услуги.

ИНТЕР РАО ЕЭС

энергия без границ

02. ПЛАТФОРМА АВТОМАТИЗАЦИИ

Инфраструктурные и процедурные меры, которые снимают барьеры и обеспечивают контроль



Markdown + Mermaid.js: прощай формализм, привет скорость

СЕЙЧАС

Word + Visio = бюрократия

Проверка отступов, шрифтов, форматирования. Красивые схемы в Visio, которые никто не читает. Технический писатель как узкое горло. Этапы согласования затягиваются на недели.

ПРЕДЛАГАЕМ

Markdown + Mermaid.js + ИИ

Только содержание, никакого форматирования. Схемы как код — версионируемые, диффабельные. LLM готовит документацию и схемы. Техписатель выходит из критического пути.

Три преимущества подхода

Скорость

Исчезает этап проверки форматирования. Согласование в дни, а не недели.

ИИ-нативность

Markdown и Mermaid — родные форматы LLM. Генерация, анализ и верификация ИИ.

Версионируемость

Схемы как код — diff, merge, code review. Никакого "потерянного Visio".

Автономность

Разработчик готовит документацию сам. Техписатель — не bottleneck.

РАЗРАБОТЧИК МАЛОЙ АВТОМАТИЗАЦИИ

- Исходный код приложения
- Документация в Markdown
- Схемы в Mermaid.js
- Dockerfile и зависимости
- Результаты автотестов

Блок ИТ (ДЕРЖАТЕЛЬ УСЛУГИ)

- VLAN и сетевое оборудование
- Серверы, IPMI, коммутаторы
- Docker-репозиторий
- Защищённые образы ОС
- Защищённые образы ОС
- GitLab-репозиторий
- SAST / SCA как сервис
- Kubernetes-платформа

Принцип: VM только по особому разрешению

По умолчанию — только среда выполнения контейнеров. Это сразу исключает из документации систему защиты контейнеризации, настройки ОС и контроль запуска — всё решается один раз на уровне платформы.

Контейнерное облако: безопасность по дизайну

Держатель инфраструктуры предоставляет среду выполнения, а не виртуальные машины

Защита контейнеризации

Защищается **один раз** при построении облака. Не требует документирования для каждого приложения.

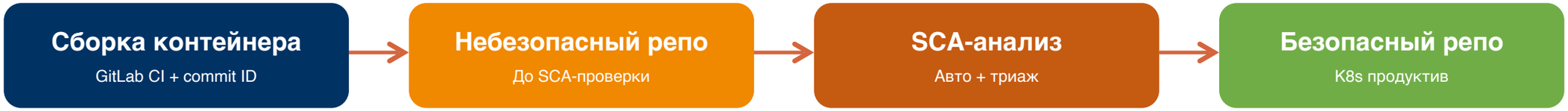
Защита и обновление ОС

Hardened-образы ОС в актуальном состоянии. Обновления доставляются **централизованно**.

Контроль запуска

Только контейнеры из корп. Docker-репозитория, **прошедшие SCA**. Никакого стороннего кода.

Поток контейнера от сборки до продуктива



Результат: разработчик фокусируется на функционале, ИБ — на инфраструктуре. Никаких VM по умолчанию — только контейнеры из проверенного репозитория в защищённом облаке Kubernetes.

Три шаблонных проекта: стартовая точка для любой автоматизации

Безопасный каркас с аутентификацией, авторизацией, ролевой моделью, логированием, 2FA и автотестами по **Методике ИБ**

Web UI

React + Python + ORM + БД

- ✓ Готовая админка пользователей
- ✓ Базовый Layout
- ✓ Аутентификация + 2FA
- ✓ Ролевая модель
- ✓ Автотесты Методики ИБ
- ✓ SAST + SCA + пентест

REST API

Python Flask + токены

- ✓ Готовая аутентификация
- ✓ JWT-токены
- ✓ Swagger-документация
- ✓ Логирование запросов
- ✓ Автотесты Методики ИБ
- ✓ SAST + SCA + пентест

MCP API

Model Context Protocol

- ✓ Подключение к ИИ-агенту
- ✓ Инструменты для LLM
- ✓ Безопасный контекст
- ✓ Аудит действий
- ✓ Автотесты Методики ИБ
- ✓ SAST + SCA + пентест

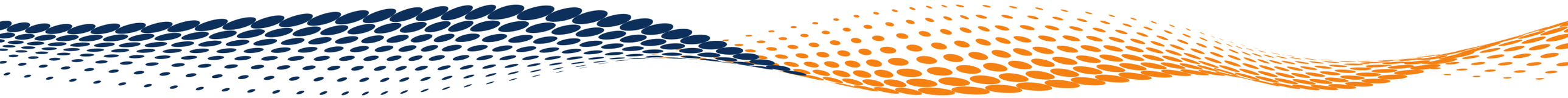
Разработчик берёт шаблон и **добавляет функционал**, не меняя безопасный каркас. Документация шаблонная — достаточно скорректировать тесты. Для сдачи ПСИ по Методике ИБ предъявляются **результаты автотестов**.

ИНТЕР РАО ЕЭС

энергия без границ

03. БЕЗОПАСНОСТЬ КАК SELF-SERVICE

Автоматизированный конвейер безопасной разработки,
работающий без ручных этапов



Централизованный GitLab на базе Блок ИТ

Единое место для всех исходных кодов малой автоматизации

Что получаем

- ✓ Увольнение ≠ потере знаний — код в компании
- ✓ Коллективное владение и code review
- ✓ Версионирование и откат изменений
- ✓ Автоподключение SAST при создании репо
- ✓ Баги высокого уровня → issues в GitLab

Как это работает

1. Разработчик создаёт репозиторий
2. Автоматическое подключение SAST
3. Каждый коммит → анализ кода
4. Триаж — централизованная услуга
5. Результаты → issues в репозитории

Снимаем ключевой риск

Увольнение сотрудника больше не означает потерю знаний. Код, документация и история изменений остаются в компании.

1 SAST

При создании репозитория каждый коммит автоматически проходит статический анализ кода.

Триаж (проверка на false positive) — централизованная услуга.

2 SCA контейнеров

В метаданных контейнера: Docker-репозиторий, список GitLab-репо и **commit ID** для верификации.

Анализ зависимостей и образов — автоматически.

3 Два Docker-репозитория

Репо 1 — образы до SCA-проверки (unsafe).

Репо 2 — образы после прохождения SCA (safe).

В продуктив — только из safe-репо.

Полный цикл без участия разработчика

Commit в GitLab → **SAST** анализ кода → **SCA** анализ контейнера → **Проход** в safe-репо → K8s

При непрохождении — автоматические issues в GitLab, уведомление разработчика, блокировка деплоя

ИНТЕР РАО ЕЭС

энергия без границ

04. ДОРОЖНАЯ КАРТА 2027–2029

Пошаговый план внедрения: от подготовки инфраструктуры до масштабирования

Три года — три фазы: фундамент, платформа, масштаб

2027

Фундамент

- ВНД на Markdown
- Бюджетирование
- K8s-лаборатория
- Регламент разделения
- Подразделение ИБ-разработки
- GitLab + Docker репо

2028

Платформа

- Запуск K8s-облака
- Шаблонные проекты
- GitLab + Docker в проде
- Защищённые образы ОС
- Автоматизация SAST/SCA
- Обновление ВНД

2029

Масштаб

- Обучающие вебинары
- Расширенная автоматизация
- Культура citizen-dev
- Массовое внедрение

Инвестиционная логика:

2027-й — бюджетирование и пилоты, 2028-й — реализация платформы, 2029-й — получение эффекта от масштабирования. Каждый год строится на результатах предыдущего.

2027: Закладываем фундамент — ВНД, бюджеты, пилоты

Нормативная база и регламенты

- Корректировка ВНД — переход с Word на Markdown, с Visio на Mermaid.js
- Регламент исключения из документов разработчика всего, что в зоне Блок ИТ
- Бюджетирование встроенного редактора Markdown в СЭД
- Разработка шаблонных проектов с автотестами Методики ИБ
- Создание подразделения

Инфраструктура

- Лаборатория по выбору платформы Kubernetes
- Закупка лицензий и серверов для частного облака
- Защита облака по ИБ
- Создание GitLab-репозитория
- Два Docker-репозитория (safe / unsafe)
- Защищённые образы ОС
- Автоматизация SAST/SCA

2028: Запускаем платформу — облако, шаблоны, процессы

Частное облако Kubernetes

Реализация продуктивной среды контейнеризации. Защита по ИБ. Мониторинг и логирование.

Шаблонные проекты

3 готовых шаблона: Web UI, REST API, MCP API. Готовый ИБ-каркас, автотесты Методики ИБ, документация.

Инфраструктура в проде

GitLab и Docker-репозитории. Защищённые образы ОС. Автоматизация SAST/SCA. Конвейер готов.

Корректировка ВНД: полный цикл внедрения

Обновлённые ВНД фиксируют полный цикл внедрения малой автоматизации:

- Исходники хранятся в GitLab-репозитории Блок ИТ
- Уязвимости SAST отрабатываются через issues в GitLab
- Docker-контейнер разворачивается из безопасного репозитория
- Доступы выдаются через централизованный процесс
- Документация готовится в Markdown + Mermaid.js

2029: Масштабируем — обучение, вебинары, культура

Обучающие вебинары

По разработке и внедрению малой автоматизации. Подключение к ИИ-агентам. Работа с шаблонными проектами. Markdown и Mermaid.js для документирования.

Расширенная автоматизация

Упрощённый конвейер от разработки до продуктива. Масштабирование платформы. Подключение новых команд и бизнес-направлений.

Целевое состояние: культура citizen-разработки

- Каждый бизнес-пользователь может быстро создавать автоматизации
- ИТ-службы сохраняют полный контроль над инфраструктурой и ИБ
- On-premise инфраструктура полностью соответствует ВНД
- Операционные затраты сокращены за счёт массовой автоматизации
- Знания сохранены в компании независимо от кадровых изменений

Синергия эффектов: что получает компания

Для бизнеса

Резкое сокращение операционных затрат. Автоматизация рутины. Быстрая реакция на изменения.

Для ИТ

Контролируемая среда. Централизованное управление. Отсутствие теневых решений.

Для ИБ

Единые стандарты. Автоматизированные проверки. Соответствие ВНД.

Для разработчиков

Низкий порог входа. Готовые шаблоны. Отсутствие бюрократии.

Для компании в целом

- Сохранение знаний — независимость от кадровых изменений
- Инвестиции в активы, а не в зависимости от конкретных людей
- Контролируемая среда без торможения инноваций
- Готовность к масштабированию без перестройки процессов

ИНТЕР РАО ЕЭС

энергия без границ

СПАСИБО ЗА ВНИМАНИЕ!

Малая автоматизация — это не угроза контролю, а возможность,
если инфраструктура готова

Вопросы для обсуждения

Блок ИТ | Июль 2026