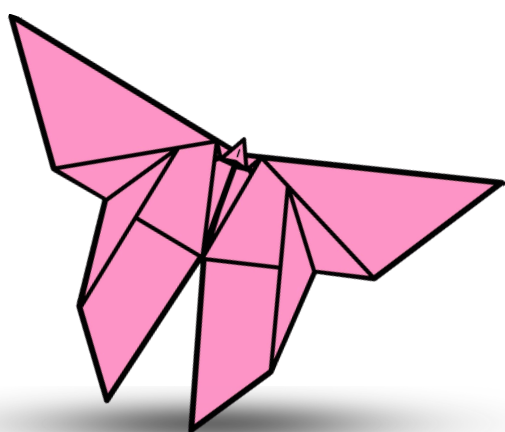


Структурная целостность JATS: Фреймворк XML-First

Архитектурные основы редакционных систем на базе JATS: комплексная модель научного производства XML-First и машиночитаемых знаний



Впервые опубликовано: Zeba Academy и Zeba Books.

Год издания: 2026

Серия: Zeba Academy Blueprints -- Технические директивы суверенных систем (Sovereign Systems Technical Directives)

Цель серии: Этот цикл директив разработан для борьбы с «эншитификацией» и избыточной перегруженностью современного ПО. Наша цель – вернуть суверенный контроль над нашими системами, сократив разрыв между глубокой академической теорией и критически важной промышленной реализацией. Мы убеждены, что программное обеспечение должно быть быстрым, долговечным и, прежде всего, понятным для его владельца и пользователя.

Главный архитектор: Суфян бин Узайр, сертифицированный Google Cloud Professional DevOps-инженер.

Основной стек: Linux, Rust, Zig, C++, Flutter и PHP.

Лицензирование и интеллектуальная собственность: Материал доступен на условиях лицензии Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0).

- **Разрешения:** Вы можете свободно распространять и адаптировать данный материал в любых целях при условии указания авторства и сохранения аналогичной лицензии для производных работ.
- **Полный текст лицензии:** <https://creativecommons.org/licenses/by-sa/4.0/>
- **Суверенная целостность:** Документ курируется человеком для исключения алгоритмического «шума». Несмотря на использование нейросетей для синтеза, каждая строка проходит проверку на соответствие стандартам высокой информативности и практической ценности.

Email: hello@zeba.academy

Структурная целостность JATS:

Фреймворк XML-First

Архитектурные основы редакционных систем на базе JATS:

комплексная модель научного производства XML-First и

машиночитаемых знаний

Философия «Единого источника истины»

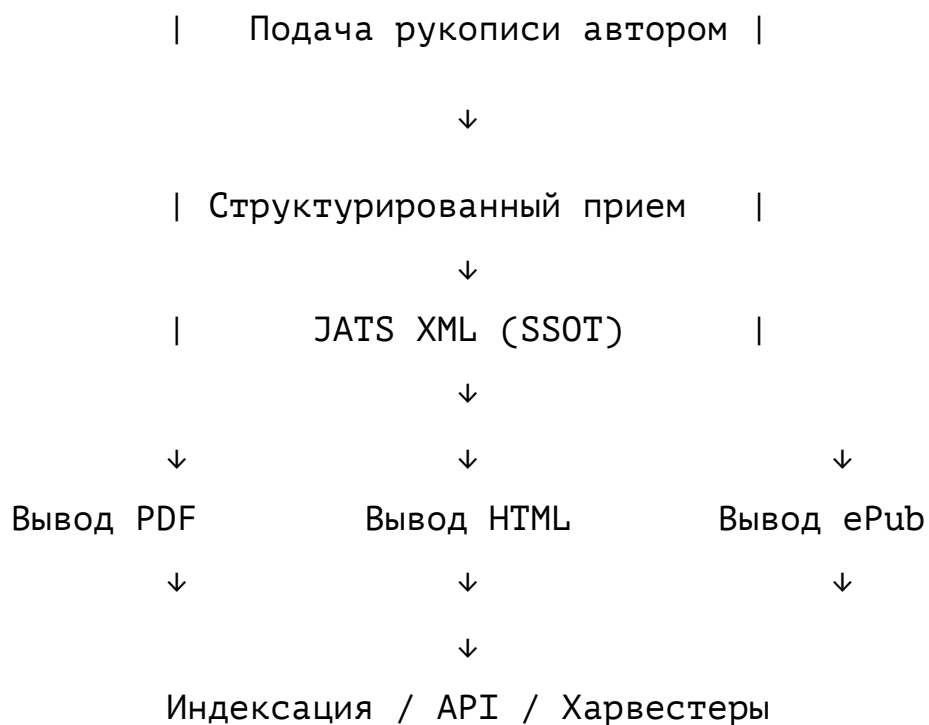
Эволюция инфраструктуры научных публикаций выявила фундаментальную неэффективность традиционных рабочих процессов: отсутствие канонического машиночитаемого представления контента. Исторически сложилось так, что издательские конвейеры рассматривали документы, преимущественно в формате PDF, как авторитетный источник. Этот подход в корне ошибочен, поскольку PDF является артефактом визуального представления, а не контейнером структурированных данных.

Философия «**Единого источника истины**» (**Single Source of Truth, SSOT**) решает эту проблему, утверждая **JATS XML (NISO Z39.96)** в качестве канонического представления всего научного контента. Каждый процесс трансформации, распространения и интеграции производят именно из этого единого структурированного источника.

Концептуальная модель

В системах, управляемых по принципу SSOT, все производные выходные форматы - PDF, HTML, ePub, индексы для агрегаторов, генерируются из

одного экземпляра XML. Это исключает несоответствия и гарантирует детерминированность выходных данных.¹



Операционные последствия

1. **Устранение избыточных трансформаций** Отпадает необходимость в повторных конвертациях между форматами. XML генерируется один раз и используется многократно.
2. **Детерминированные конвейеры рендеринга** Движки визуализации выдают согласованные результаты, так как входные данные стандартизированы.

¹ Conceptual Model (SSOT / Single-Source Publishing) - https://en.wikipedia.org/wiki/Single-source_publishing (en.wikipedia.org) - Дата обращения: 19 марта 2026 г

3. **Централизация метаданных** Все метаданные: аффилиации авторов, пристатейные списки литературы, сведения о финансировании хранятся в рамках единой схемы.²
4. **Аудируемость и прослеживаемость** Контроль версий может применяться на уровне XML, что обеспечивает точное отслеживание всех внесенных изменений.

Требования к внедрению SSOT

Для успешной реализации модели «Единого источника истины» необходимо:

- Обеспечить генерацию XML **на самом раннем этапе приема контента.**
- Исключить использование PDF в качестве авторитетного источника.
- Внедрить контрольные **точки валидации по схеме.**
- Обеспечить **двустороннюю синхронизацию** между редакционными системами и XML.

Проектировочные ограничения

- XML должен строго соответствовать **JATS DTD или XSD.**
- Все редакционные действия должны **учитывать структуру схемы (schema-aware).**
- Слои рендеринга должны быть **без сохранения состояния (stateless) и идемпотентными.**

² Operational Implications & XML Schema Enforcement - <https://www.w3.org/TR/xmlschema-1/> - Дата обращения: 19 марта 2026 г

XML как императив: почему PDF-центричные рабочие процессы препятствуют современному научному поиску

Сохранение PDF-центричных рабочих процессов обусловлено скорее исторической инерцией, чем технической целесообразностью. Хотя формат PDF обеспечивает визуальную точность, он фундаментально не отвечает требованиям современных цифровых экосистем, особенно тех, что основаны на машинном обучении, семантическом индексировании и обмене данными через API.³

Структурные ограничения PDF

PDF кодирует контент как:

- Позиционированные глифы (символы)
- Инструкции по разметке страницы
- Встроенные шрифты

При этом PDF не кодирует:

- Логическую структуру (разделы, абзацы)
- Семантическое значение (роли авторов, цитирование)
- Машиночитаемые метаданные

Сложность извлечения данных

Инструменты вроде GROBID пытаются реконструировать структуру из PDF, используя вероятностные модели. Однако:

- Точность варьируется в зависимости от сложности верстки.

³ XML vs PDF in Scholarly Publishing (JATS & Machine-Readable Advantage) - <https://jats.nlm.nih.gov/> - Дата обращения: 19 марта 2026 г

- Таблицы и уравнения часто повреждаются при извлечении.
- Парсинг пристатейных списков литературы подвержен ошибкам.

Это вносит фактор **недетерминированности**, что недопустимо в производственных конвейерах промышленного уровня.

Фрагментация метаданных

В традиционных рабочих процессах метаданные распределены по нескольким уровням:

Уровень	Источник
Система подачи (Submission)	Ввод данных пользователем
PDF	Внедренные или извлеченные данные
Фиды для индексации	Реконструированные данные

Это приводит к:

- Несогласованности имен авторов
- Отсутствию идентификаторов ORCID
- Битым ссылкам в списках цитирования

JATS XML как решение

JATS (Journal Article Tag Suite) устраняет эти проблемы, обеспечивая:

- Явную семантическую разметку
- Иерархическую структуру
- Стандартизированные поля метаданных

Пример семантической полноты:

```
<contrib contrib-type="author">
  <name>
    <surname>Smith</surname>
    <given-names>Jane</given-names>
  </name>
  <contrib-id
contrib-id-type="orcid">0000-0002-1825-0097</contrib-id>
</contrib>
```

Это позволяет реализовать:

- Бесшовную интеграцию с ORCID
- Устранение неоднозначности имен авторов
- Автоматизированную индексацию

Преимущества для поиска и индексации

JATS XML обеспечивает прямой импорт данных в:

- Регистраторы DOI (Crossref)
- Репозитории PubMed Central
- Семантические поисковые системы

Готовность к машинному обучению

Структурированный XML обеспечивает поддержку:

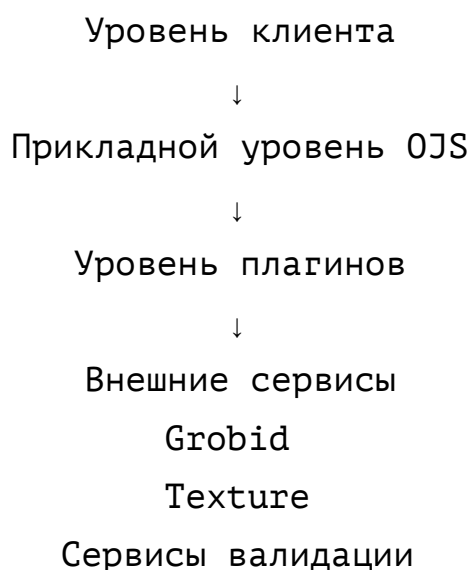
- Распознавания именованных сущностей (NER)
- Построения графов цитирования
- Интеграции в графы знаний

Основной технологический стек

OJS 3.x как уровень оркестрации

Open Journal Systems (OJS 3.x) функционирует как **центральный уровень оркестрации**, координируя полный жизненный цикл рукописи и выступая в роли плоскости управления (control plane) для всех последующих XML-центричных процессов. Система обеспечивает переходы состояний рабочего процесса, нормализацию метаданных и инициирует триггеры интеграции с внешними сервисами.⁴

Архитектура системы



Ключевые обязанности

Управление подачей рукописей

- Прием файлов в различных форматах (DOCX, PDF, LaTeX)
- Структурированный захват метаданных (ORCID, аффилиации, данные о финансировании)

⁴ Open Journal Systems (OJS) Workflow & Architecture - https://en.wikipedia.org/wiki/Open_Journal_Systems - Дата обращения: 19 марта 2026 г

- Валидация и нормализация ролей авторов

Редакционный рабочий процесс

- Оркестрация асинхронного рецензирования
- Версионирование правок с ведением журналов аудита
- Управление состояниями принятия решений через этапы рабочего процесса

Производственная интеграция

- Контроль исполнения конвейера по приему данных в формате XML-first
- Синхронизация между метаданными базы данных и узлами JATS
- Хранение XML-артефактов с поддержкой контроля версий

Расширяемость на основе плагинов

OJS обеспечивает расширяемость за счет модульных плагинов:

- Плагин импорта/экспорта XML (прием и экспорт JATS)
- Плагин REST API (программное взаимодействие)
- Интерфейс OAI-PMH (сбор метаданных)

Пользовательские плагины могут внедрять хуки валидации или инициировать внешние конвейеры обработки во время событий подачи или публикации.

Интеграция Grobid

GROBID функционирует как **микросервис приема данных без сохранения состояния (stateless)**, преобразуя неструктурированные PDF-файлы в формат TEI XML с использованием моделей машинного обучения для сегментации документов и парсинга цитирований.

Интеграция в конвейер

PDF-файл → Grobid → TEI XML → XSLT → JATS XML

Масштабирование Grobid

- Контейнеризированное развертывание обеспечивает горизонтальную масштабируемость.
- Балансировка нагрузки между несколькими экземплярами для пакетной обработки данных.

```
docker run -p 8070:8070 lfoppiano/grobid:latest
```

Пакетная обработка

```
for file in *.pdf; do  
  curl -F "input=@$file"  
  http://localhost:8070/api/processFulltextDocument  
done
```

XML-редактор Texture

Texture предоставляет браузерную среду редактирования с поддержкой схем для ручной валидации и корректировки XML, созданного в автоматическом режиме.

Возможности

- Визуальное структурное редактирование, привязанное к узлам XML.
- Встроенная валидация схемы (соответствие стандарту JATS).
- Мгновенная обратная связь при возникновении структурных несоответствий.

Интеграция в рабочий процесс

- Загрузка JATS XML, полученного на этапе трансформации.
- Выполнение семантических правок (списки литературы, разделы, метаданные).
- Валидация на соответствие ограничениям схемы.
- Экспорт нормализованного XML обратно в производственный конвейер OJS.

Уровень трансформации (TEI → JATS)

Уровень трансформации обеспечивает стандартизацию разнородных XML-структур в формат, **соответствующий стандарту JATS**, с использованием конвейеров на базе XSLT.

Пример сопоставления (Mapping)

```
<xsl:template match="tei:p">
  <p><xsl:apply-templates/></p>
</xsl:template>
```

Лучшие практики

- Модульный дизайн XSLT, разделяющий обработку метаданных, основного текста и списков литературы.
- Нормализация пространств имен для предотвращения конфликтов схем.
- Надежная обработка ошибок для отсутствующих или некорректно сформированных узлов.
- Использование предкомпилированных процессоров XSLT (например, Saxon) для оптимизации производительности.

Этот уровень гарантирует детерминированную и воспроизводимую конвертацию, формируя основу издательского конвейера JATS-first.

Производственный конвейер

Прием препринтов (Pre-Print Intake)

Уровень приема отвечает за нормализацию разнородных авторских материалов в структурированную, машиночитаемую промежуточную форму, обеспечивая согласованность данных перед их преобразованием в JATS XML. На этом этапе необходимо обеспечить стандартизацию форматов, нормализацию кодировки (UTF-8) и извлечение метаданных.⁵

Формат Инструмент

DOCX Pandoc

LaTeX LaTeXML

PDF Grobid

Конвертация через Pandoc

```
pandoc input.docx -t jats -o output.xml
```

Технические аспекты

- Нормализация стилей (Заголовок 1 → <sec>) перед конвертацией.
- Очистка от артефактов встроенного форматирования (шрифты, цвета).
- Извлечение внедренных метаданных (авторы, ключевые слова).
- Валидация кодировки и наборов символов.

⁵ Pre-Print Intake, JATS Conversion & Validation Pipeline - <https://en.wikipedia.org/wiki/Pandoc> - Дата обращения: 19 марта 2026 г

Структурная разметка

После конвертации документы должны пройти этап **семантического структурирования**, обеспечивающего строгое соблюдение иерархии JATS и ограничений для элементов.

Обеспечение базовой структуры

- Иерархия разделов → <sec> с вложенными <title>
- Рисунки → <fig> с <caption> и <graphic>
- Таблицы → <table-wrap> со структурированным <tbody>

Правила автоматической разметки

- Распознавание заголовков → <title> внутри <sec>
- Группировка абзацев → <p> с логической сегментацией
- Цитирования → <xref ref-type="bibr">, связанные с <ref-list>

Расширенные улучшения разметки

- Нормализация ORCID в <contrib-id>
- Метаданные о финансировании → <funding-group>
- Внутрискочные формулы → внедрение MathML

Циклы валидации (Validator Loops)

Валидация гарантирует строгое соответствие стандарту NISO Z39.96 (JATS) и предотвращает отклонение от схемы в процессе трансформации и редактирования.

Стек валидации

- xmllint → проверка синтаксиса
- Схемы RELAX NG → структурная валидация

- Рекомендации JATS4R → контроль соблюдения лучших практик

Модель непрерывной валидации

Ввод → Валидация → Ошибка → Исправление → Повторная валидация → Одобрение

Интеграция в CI/CD

Валидация должна быть автоматизирована внутри конвейеров:

```
xmllint --noout article.xml || exit 1
```

Технические улучшения

- Pre-commit хуки для валидации XML.
- Автоматическая отчетность об ошибках с диагностикой на уровне строк.
- Интеграция с Texture для рабочих процессов внесения исправлений.

Мультиформатный рендеринг

Рендеринг преобразует валидированный JATS XML в готовые к распространению форматы посредством детерминированных преобразований без сохранения состояния (stateless).

Генерация PDF

- Конвейер трансформации XSL-FO
- Визуализация через Apache FOP

Вывод HTML

```
xsltproc jats-html.xsl article.xml
```

Упаковка ePub

- HTML + OPF (метаданные) + NCX (навигация)

- Упаковка в ZIP-совместимый контейнер EPUB

Оптимизация рендеринга

- Предкомпилированные таблицы стилей XSLT для снижения задержек
- Параллельные конвейеры рендеринга для пакетной обработки
- Распространение статических ресурсов через CDN

Вопросы масштабируемости

- Микросервисы рендеринга без сохранения состояния
- Планирование задач на основе очередей (например, RabbitMQ)
- Кэширование результатов рендеринга для минимизации повторных вычислений

Этот конвейер обеспечивает **высокую пропускную способность, воспроизводимость и соответствие стандартов** выпускаемой издательской продукции, формируя операционную основу системы JATS-first.

Протоколы взаимодействия

REST API

Open Journal Systems предоставляет уровень RESTful API, который обеспечивает программный доступ к поданным рукописям, метаданным и издательским рабочим процессам. Этот API служит основным интерфейсом для интеграции внешних систем в конвейер JATS-first.⁶

Основные эндпоинты

`GET /api/v1/articles`

⁶ Interoperability Protocols (OAI-PMH Standard) - <https://www.openarchives.org/OAI/openarchivesprotocol.html> - Дата обращения: 19 марта 2026 г

POST /api/v1/submissions

Аутентификация

- **JWT (JSON Web Tokens)** для аутентификации без сохранения состояния (stateless).
- **OAuth2** для делегированной авторизации и интеграции со сторонними сервисами.

Технические возможности

- CRUD-операции для заявок и статей.
- Извлечение метаданных в структурированных форматах (JSON/XML).
- Управление переходами состояний рабочего процесса через API-триггеры.

Сценарии интеграции

- Синхронизация с внешними CMS (headless-публикация).
- Конвейеры аналитики (метрики использования, отслеживание цитирований).
- Системы индексации на базе ИИ (семантический парсинг, графы знаний).

Лучшие практики

- Ограничение частоты запросов (rate limiting) для стабильности API.
- Версионирование эндпоинтов для обеспечения обратной совместимости.
- Безопасная передача данных через HTTPS с политиками истечения срока действия токенов.

OAI-PMH

OAI-PMH предоставляет стандартизированный протокол сбора (harvesting) для передачи структурированных метаданных агрегаторам и сервисам индексирования.⁷

Поддерживаемые методы (Verbs)

- **Identify** → метаданные репозитория.
- **ListRecords** → массовый сбор метаданных.
- **GetRecord** → извлечение отдельной записи.

Пример запроса

```
?verb>ListRecords&metadataPrefix=jats
```

Форматы метаданных

- **Dublin Core (oai_dc)** для базовой совместимости.
- **JATS XML (jats)** для передачи всей полноты семантических данных.

Технические характеристики

- Инкрементальный сбор данных с использованием временных меток (datestamps).
- Пагинация с помощью токенов возобновления (resumption tokens).
- Модель «запрос-ответ» без сохранения состояния.

Стратегии оптимизации

- Предварительная генерация кэша метаданных для ускорения ответов.
- Сжатие ответов (gzip) для больших наборов данных.

⁷ OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting) - <https://www.openarchives.org/OAI/openarchivesprotocol.html> - Дата обращения: 19 марта 2026 г

- Обеспечение согласованного сопоставления идентификаторов (OAI identifiers ↔ DOI).

Системы Crossref и DOI

JATS XML программно преобразуется в схему депозита Crossref, что обеспечивает регистрацию DOI и глобальную находимость (discoverability) контента.⁸

Автоматизированный рабочий процесс DOI

1. Извлечение структурированных метаданных из JATS XML.
2. Трансформация в XML, соответствующий требованиям Crossref (через XSLT).
3. Отправка депозита через Crossref API.
4. Получение и регистрация DOI.

Пример потока депозита:

JATS XML → XSLT → Crossref XML → Отправка через API →
Присвоение DOI

Технические аспекты

- Обязательные поля: заголовок, авторы, дата публикации.
- Связывание ссылок для интеграции в сеть цитирования.
- Обогащение метаданных идентификаторами ORCID и данными о финансировании.

Обработка ошибок

- Ошибки валидации схемы (отклонение XML).

⁸ Crossref DOI Registration & Metadata Deposit - <https://www.crossref.org/documentation/> - Дата обращения: 19 марта 2026 г

- Обнаружение дубликатов DOI и разрешение конфликтов.
- Логирующие несоответствий метаданных между OJS и Crossref.

Повышение надежности

- Механизмы повторных попыток для неудачных депозитов.
- Асинхронные очереди обработки.
- Журналы аудита для отслеживания жизненного цикла DOI.

Данный уровень взаимодействия обеспечивает бесшовную интеграцию с глобальной научной инфраструктурой, позволяя осуществлять обмен метаданными **в реальном времени, автоматизированную индексацию и персистентную идентификацию**, что критически важно для масштабируемых машиночитаемых издательских экосистем.

Метрики устойчивости

Модель снижения затрат

Архитектура JATS-first внедряет **системную оптимизацию затрат** за счет устранения избыточных трансформаций и минимизации ручного вмешательства на протяжении всего производственного цикла. Автоматизация заменяет трудоемкие этапы, такие как верстка, конвертация форматов и ручная корректировка метаданных.⁹

Автоматизация сокращает:

- Ручную верстку благодаря генерации PDF на основе XSL-FO.
- Повторяющиеся конвертации форматов через рендеринг из единого XML-источника.

⁹ Event-Driven Architecture & Scalable Systems - <https://microservices.io/patterns/data/event-driven-architecture.html> - Дата обращения: 19 марта 2026 г

- Человеческий фактор благодаря валидации на основе схем и автоматизированным конвейерам.

Количественные показатели

Метрика	Традиционный подход	JATS-first
Время обработки	5–10 дней	1–2 дня
Стоимость статьи	Высокая	Сниженная
Частота ошибок	Высокая	Минимальная

Технический вывод:

Снижение затрат достигается путем перехода от процессов, **зависящих от человека, к детерминированным машинным конвейерам**, где XML выступает в качестве многоразового актива для всех выходных форматов.¹⁰

Стратегия автоматизации

Автоматизация реализуется через **распределенную событийно-ориентированную архитектуру**, обеспечивающую масштабируемость и отказоустойчивость.

Основные компоненты

- **Обработка на основе очередей** (RabbitMQ/Kafka) для асинхронного выполнения задач.
- **Оркестрация контейнеров** (Docker/Kubernetes) для изоляции и масштабирования сервисов.

¹⁰ Deterministic Systems & Automation (CI/CD Principles) - <https://martinfowler.com/articles/continuousIntegration.html> - Дата обращения: 19 марта 2026 г

- **Микросервисы без сохранения состояния** для парсинга, валидации и рендеринга.

Модель конвейера:

Подача → Очередь → Рабочие узлы (Worker Nodes) →
Обработка XML → Вывод / Дистрибуция

Технические преимущества

- Децентрализованные сервисы устраняют «узкие места» системы.
- Механизмы повторного выполнения для сбойных задач.
- Горизонтальная масштабируемость при переменных нагрузках.

Мониторинг и KPI

Операционная устойчивость требует **непрерывного наблюдения (observability) и отслеживания производительности** на всем протяжении конвейера.¹¹

Ключевые метрики

- Доля успешных валидаций XML (целевой показатель >99%)
- Сквозная задержка обработки одной статьи (latency)
- Время ответа и пропускная способность API
- Частота ошибок на уровнях парсинга и трансформации

Стек мониторинга

- Prometheus (сбор метрик)
- Grafana (дашборды визуализации)
- Централизованное логирование (стек ELK)

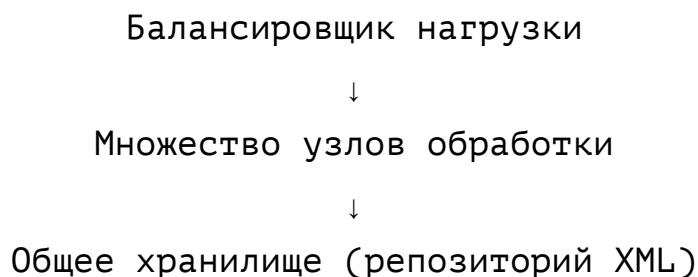
¹¹ Monitoring & Observability (Metrics and KPIs) - <https://prometheus.io/docs/introduction/overview/> -
Дата обращения: 19 марта 2026 г

Механизмы оповещения

- Оповещения на основе пороговых значений для сбоев валидации
- Мониторинг бэклога
- Проверка работоспособности сервисов через эндпоинты heartbeat

Модель масштабирования

Архитектура поддерживает **горизонтальное масштабирование** для обработки растущих объемов заявок без потери производительности.¹²



Детали реализации

- Сервисы без сохранения состояния (stateless) позволяют выполнять динамическое масштабирование.
- Распределенное хранилище (S3/MinIO) для персистентности XML.
- Балансировка нагрузки обеспечивает равномерное распределение задач.

Долгосрочная устойчивость

Устойчивость обеспечивается за счет проектирования с поддержкой прямой совместимости и модульной расширяемости.

¹² Horizontal Scalability & Load-Balanced Architecture - <https://en.wikipedia.org/wiki/Scalability>
- Дата обращения: 19 марта 2026 г

Ключевые принципы

- Поддержка эволюции схем (обновления JATS без остановки конвейера)
- Обратная совместимость для устаревших версий XML
- Модульная модернизация конвейера без глобального рефакторинга всей системы

Стратегический результат

Этот подход гарантирует, что издательская инфраструктура остается **жизнеспособной, адаптивной и ориентированной на будущее**,¹³ способной интегрировать новые стандарты, инструменты и требования к взаимодействию без значительных архитектурных изменений.

Заключение

Архитектура JATS-first представляет собой фундаментальную трансформацию систем научных публикаций. Смещая каноническое представление знаний от статических документов к структурированному машиночитаемому XML, организации могут достичь беспрецедентной эффективности в производстве, распространении и поиске контента.¹⁴

Данная концепция демонстрирует, как реализовать эту парадигму на практике через:

- Структурированные конвейеры приема данных
- Автоматизированные уровни трансформации
- Непрерывные циклы валидации
- Интероперабельные API и протоколы

¹³ Future-Proof & Scalable Architecture (Microservices Principles) - <https://martinfowler.com/microservices/> - Дата обращения: 19 марта 2026 г

¹⁴ JATS XML in Scholarly Publishing - <https://jats.nlm.nih.gov/> - Дата обращения: 19 марта 2026 г

Для системных специалистов ключевой вывод заключается в том, что **XML - это не просто формат, это инфраструктурный слой**, на котором должна строиться современная научная коммуникация.

Zeba Academy - это специализированная инициатива в области технических исследований и обучения, основанная на принципах суверенной системной инженерии. Проект, созданный Суфьяном бин Узайром - автором, университетским преподавателем и сертифицированным Google Cloud DevOps-инженером, служит мостом между академической теорией и реализацией критически важных систем.

Мы отвергаем «эншитификацию» современного ПО. Наша основная миссия - продвижение **архитектуры без балласта (Anti-Bloat Architecture)** через освоение следующих направлений:

- **Системные языки.** Разработка на Rust, Zig и C++ высокопроизводительных фундаментов с упором на безопасность памяти и детерминизм исполнения.
- **SRE и DevOps.** Автоматизация профессионального уровня на базе Google Cloud, Terraform и неизменяемой инфраструктуры (Immutable Infrastructure). Мы ликвидируем операционную хрупкость и ручной труд (toil).
- **Высокопроизводительные интерфейсы.** Проектирование на Flutter кроссплатформенных систем с нативным откликом. Без компромиссов и задержек, присущих стандартным веб-оболочкам.
- **Регенерация веб-издательства.** Возврат WordPress и PHP в строй через радикальную очистку от «шлака». Объектное кэширование в Redis и Unix-сокеты превращают стандартные платформы в скоростные геостабильные движки.
- **Модернизация Legacy-систем.** Перенос классических вычислительных задач и старых кодовых баз на C в современные парадигмы безопасной работы с памятью и актуальные системы сборки.

Zeba Academy не просто учит писать код - мы проектируем надежность. Объединяя аналитическую строгость исторических исследований с точностью сертифицированной облачной инженерии Google, мы вооружаем наших «оперативников» директивами, необходимыми для создания систем, которые будут безопасными, быстрыми и долговечными.

Вебсайт: - <https://zeba.academy>



Zeba Academy

zeba.academy
