

УДК 004.9 DOI: 10.14451/1.259.133

Экономическая оценка архитектурных методов интеграции данных в цифровых двойниках

© 2026 Яшина Виктория Денисовна

Преподаватель. Нижегородский государственный педагогический университет им. К. Минина.
E-mail: mvitalinaya15@yandex.ru

© 2026 Круглов Илья Александрович

Преподаватель. Нижегородский государственный педагогический университет им. К. Минина.
E-mail: kruglov@volga-shipyard.ru

© 2026 Поначугин Александр Викторович

Заведующий кафедрой, Доцент, Кандидат экономических наук. Нижегородский государственный педагогический университет им. К. Минина.
E-mail: sasha3@bk.ru

Ключевые слова: цифровой двойник, интеграция данных, IoT, потоковая обработка, брокер сообщений, Kafka, классификация архитектур, экономическая эффективность.

В статье рассматриваются архитектурные методы интеграции данных в цифровых двойниках. Проведен анализ потокового, периферийного и гибридного подходов к передаче данных от IoT-устройств. Предложена авторская классификация методов интеграции на основе трех критериев: способа передачи, типа хранения данных и топологии сети. Сформулированы практические рекомендации по выбору архитектуры в зависимости от требований к латентности и масштабируемости, а также дана экономическая оценка различных вариантов построения систем интеграции.

В условиях цифровой трансформации промышленности технология цифровых двойников (digital twins) становится ключевым инструментом управления сложными киберфизическими системами. Согласно данным Grand View Research, мировой рынок цифровых двойников вырастет с 3,76 млрд долларов США в 2018 году до 57,38 млрд долларов к 2027 году. В России, по данным НИУ ВШЭ, около 22% крупных компаний уже используют цифровые двойники, а еще 34% планируют их внедрение в ближайшей перспективе.

Понятие цифрового двойника впервые предложено профессором Мичиганского университета Майклом Гривсом в 2002 году для описания виртуального представления физического продукта. Однако, как отмечают авторы обзорных исследований [5; 10], до сих пор отсутствует единая стандартизированная архитектура, а выбор методов интеграции данных часто осуществляется эвристически. Существующие публикации фокусируются на отраслевых применениях, оставляя вне рассмотрения вопросы систематизации архитектурных подходов к интеграции данных.

С экономической точки зрения, необоснованный выбор архитектуры интеграции данных приводит к избыточным капитальным затратам на инфраструктуру и росту операционных расходов на ее сопровождение. Разработка классификации, учитывающей экономическую эффективность различных методов интеграции, позволяет предприятиям оптимизировать инвестиции при внедрении цифровых двойников и сократить совокупную стоимость владения такими системами.

Целью настоящей работы является разработка классификации архитектурных методов интеграции данных для цифровых двойников.

Для достижения поставленной цели необходимо решить следующие задачи:

- Уточнить понятийный аппарат в области цифровых двойников и интеграции данных, выделив ключевые характеристики и компоненты.
- Выявить основные архитектурные подходы к интеграции данных, используемые в современных цифровых двойниках, на основе анализа научной литературы.
- Предложить классификацию методов интеграции на основе выделенных критериев.
- Провести сравнительный анализ выделенных подходов и сформулировать рекомендации по их выбору в зависимости от требований к реальному времени и масштабируемости, а также дать экономическую оценку различным архитектурным решениям.

Научная новизна работы заключается в разработке трехуровневой классификации архитектур методов интеграции данных для цифровых двойников, учитывающей способ передачи (поточный/периферийный), тип хранения данных (нативные и надстрочные СУБД временных рядов) и топологию (брокерная/шинная/клиент-серверная), а также в выявлении критериев выбора архитектуры в зависимости от требований к латентности и масштабируемости.

Практическая значимость работы состоит в том, что предложенная классификация может быть использована при проектировании цифровых

двойников для выбора оптимального технологического стека интеграции данных, что позволит сократить время и затраты на разработку, а также обосновать инвестиции с точки зрения совокупности стоимости владения.

Под цифровым двойником (digital twin) в данной работе понимается виртуальное представление физического объекта, процесса или системы, которое находится в двусторонней связи со своим реальным прототипом, обеспечивая мониторинг, анализ и оптимизацию его функционирования.

Концептуальная модель цифрового двойника, предложенная М. Гривсом в 2002 году, включает три компонента: физический продукт, виртуальный продукт и информационную связь между ними [10, с. 20]. Ключевым отличием цифрового двойника от статической цифровой модели является наличие двусторонней обратной связи: модель не только имитирует поведение объекта, но и обновляется на основе данных с датчиков, а также может вырабатывать управляющие воздействия [5, с. 3–4].

Промышленный интернет-консорциум (IIC) определяет цифровой двойник как «формальное цифровое представление актива, процесса или системы, которое фиксирует атрибуты и поведение этого объекта, подходящие для передачи, хранения, интерпретации или обработки в определенном контексте» [2, с. 1]. Важно различать понятия «цифровой двойник» и «цифровая тень»: цифровая тень — это лишь набор данных, получаемых с датчиков, в то время как цифровой двойник предполагает наличие двусторонней связи и управляющих воздействий [8, с. 4–5].

Цифровой двойник полагается на датчики Интернета вещей (IoT) для сбора оперативных данных о физическом объекте в реальном времени [5, с. 108]. Процесс передачи, обработки и хранения этих данных составляет суть интеграции данных в контексте цифровых двойников.

В данной работе под методом интеграции данных понимается способ организации информационного обмена между физическим объектом

и его цифровым двойником, включая используемые протоколы, топологию сети, частоту передачи данных и способ хранения.

Ключевое отличие цифрового двойника от классической имитационной модели состоит в том, что изменения состояния в цифровом двойнике инициируются не статистическими распределениями, а реальными данными с датчиков. Это, в свою очередь, накладывает требования к низкой латентности, высокой пропускной способности и надежности доставки сообщений [11, с. 4].

Анализ научной литературы и отраслевых публикаций позволяет выделить три основных подхода к интеграции данных в архитектурах цифровых двойников: потоковая передача (stream integration), периферийная обработка (edge integration) и гибридные схемы, сочетающие оба метода. Каждый из подходов имеет свои преимущества, ограничения и области применения.

Потоковая интеграция предполагает непрерывную передачу данных от множества IoT-датчиков к цифровому двойнику в режиме, близком к реальному времени. Данные поступают непосредственно в виртуальную модель, что позволяет ей отражать текущее состояние физического объекта с минимальной задержкой [5, с. 109].

В основе потокового подхода лежат легковесные протоколы publish-subscribe. Наиболее распространенными являются:

- MQTT (Message Queuing Transport) – протокол, оптимизированный для устройств с ограниченными ресурсами, работающий по модели «издатель-подписчик» [5, с. 109];
- OPC UA (Open Platform Communications Unified Architecture) – промышленный стандарт, обеспечивающий интероперабельность оборудования, безопасность и расширенные возможности моделирования данных [5, с. 109];
- Apache Kafka – распределенная платформа для потоковой обработки данных с высокой пропускной способностью и низкой задержкой [6, с. 6].

Преимуществами потоковой интеграции являются максимальная актуальность данных и возможность мгновенного реагирования на отклонения в работе оборудования. К ограничениям можно отнести необходимость стабильной высокоскоростной сети, значительные вычислительные ресурсы для обработки непрерывных потоков и сложность синхронизации разнородных источников данных [5, с. 109–110].

Периферийная интеграция предполагает, что часть обработки данных выполняется непосредственно на стороне IoT-устройств или на локальных шлюзах («на краю» сети). В цифровой двойник передается уже обработанная, агрегированная и отфильтрованная информация [5, с. 110].

Такой подход позволяет снизить сетевую нагрузку и уменьшить задержки за счет локальной обработки. Он особенно эффективен в производственных сценариях с большим количеством датчиков, где сетевые каналы могут быть перегружены, или при необходимости автономной работы узлов при сбоях связи [5, с. 110]. Кроме того, на «краю» могут применяться шлюзы, объединяющие разные протоколы (например, MQTT и OPC UA) и стандартизирующие модель данных [5, с. 110].

Ограничением данного подхода является потенциальное замедление поступления части данных в центральную модель, что может быть критично для задач, требующих глобальной аналитики в реальном времени [5, с. 110].

Гибридные схемы сочетают потоковую и периферийную интерпретацию, распределяя обработку данных между уровнями «края» и облака (или центрального сервера). Локальные устройства выполняют первичную фильтрацию и агрегацию, а в центральную систему передаются либо критические события, либо данные, которые требуют комплексного анализа [6, с. 11–12].

На основе анализа литературы, представленного в разделе 3, а также с учетом требований, предъявляемых к современным цифровым двойникам, ниже предлагается авторская классификация архитектурных методов интеграции данных.

Предлагаемая классификация базируется на трех критериях, наиболее существенных для проектирования систем интеграции данных в цифровых двойниках: способ передачи данных (непрерывный поток или передача после пред-

варительной обработки), тип хранения данных (нативная СУБД временных рядов или надстроечная СУБД поверх реляционной/NoSQL-базы), а также топология сети (брокерная, шина данных или клиент-серверная).

Таблица 1. Классификация архитектур методов интеграции данных для цифровых двойников.

Критерий	Тип архитектуры	Характеристика	Примеры технологий
Способ передачи	Потоковый (Stream)	Непрерывная передача данных в реальном времени, низкая латентность.	Apache Kafka, MQTT
Способ передачи	Периферийный (Edge)	Локальная обработка на стороне датчиков, передача агрегированных данных.	IoT-шлюзы, MQTT + фильтрация
Тип хранения	Нативная СУБД-ВР	Самостоятельная система, оптимизированная под временные ряды.	InfluxDB, Prometheus
Тип хранения	Надстроечная СУБД-ВР	Реализована поверх реляционной или NoSQL-СУБД.	TimescaleDB (на PostgreSQL), OpenTSDB (на HBase)
Топология	Брокерная (Broker)	Асинхронная публикация-подписка через центральный брокер.	Apache Kafka, RabbitMQ
Топология	Шина данных (Data Bus)	Единая шина с маршрутизацией и преобразованием сообщений.	OPC UA, Apache Camel
Топология	Клиент-серверная	Синхронные запросы-ответы между компонентами.	REST API, gRPC

На основе таблицы 1 выделены четыре группы методов интеграции:

1. Группа А (брокерские потоковые, например Kafka + MQTT) обеспечивает минимальную латентность и масштабируемость, рекомендуется для систем с числом датчиков более 1000.
2. Группа Б (периферийные, например IoT-шлюзы) снижает нагрузку на сеть, рекомендуется для распределенных систем с ограниченной пропускной способностью.
3. Группа В (надстроечные СУБД-ВР, например TimescaleDB) позволяет использовать существующую инфраструктуру, рекомендуется для интеграции с legacy-системами.
4. Группа Г (гибридные) сочетает низкую латентность с хранением истории и является наиболее универсальным решением для промышленных сценариев.

На основе предложенной ранее классификации проведем сравнительный анализ выделенных групп методов интеграции по ключевым практическим параметрам: латентность, масштабируемость, сложность внедрения и пригодность для работы в реальном времени. Результаты сведены в таблицу 2.

Таблица 2. Сравнительный анализ групп методов интеграции данных.

Параметр	Брокерские потоковые	Периферийные	Надстроечные	Гибридные
Латентность	Очень низкая (<100 мс)	Низкая (после агрегации)	Высокая (зависит от запроса)	Низкая
Масштабируемость	Высокая (горизонтальная)	Средняя (ограничена шлюзом)	Средняя	Высокая
Сложность внедрения	Средняя	Низкая	Низкая (есть готовая инфраструктура)	Высокая
Подходит для реального времени	Да	Частично	Нет	Да
Подходит для хранения истории	Частично (через отдельное хранилище)	Частично	Да	Да
Требования к сети	Высокое (стабильность, скорость)	Низкие	Средние	Высокие

На основе проведенного сравнения могут быть сформулированы следующие рекомендации:

- Брокерские потоковые методы рекомендуются выбирать в сценариях, где требуется минимальная задержка передачи данных (менее 100 мс), число датчиков превышает 1000 единиц, а пропускная способность сети достаточно для непрерывной передачи потоков. Типичные применения: мониторинг быстропротекающих процессов, предиктивная аналитика состояния оборудования в реальном времени.
- Периферийные методы целесообразно применять в распределенных системах с ограниченной пропускной способностью каналов связи, а также в ситуациях, когда часть данных может быть обработана локально без потери ценности для конечного приложения. Типичные применения: удаленные промышленные объекты, автономные сенсорные сети.
- Методы на основе надстроечных СУБД-ВР оправданы в проектах, где уже развернута инфраструктура реляционных или NoSQL-баз данных, а требования к латентности не являются критическими. Также этот подход удобен для задач, связанных с анализом больших объемов исторических данных [8, с. 5–6].
- Гибридные методы представляют собой наи-

более универсальное решение и рекомендуются для большинства промышленных сценариев, где одновременно требуются низкая латентность потоковой обработки и возможность долгосрочного хранения исторических данных для последующего анализа. Данный подход сочетает достоинства брокерных потоковых методов и систем хранения временных рядов, однако требует более высоких затрат на внедрение и сопровождение.

Выбор архитектуры интеграции данных напрямую влияет на экономическую эффективность проекта цифрового двойника. Брокерские потоковые методы (группа А) требуют высоких капитальных затрат на серверное оборудование, но минимизируют простои производства, что снижает операционные расходы, и экономически оправданы для непрерывных производств с высокими требованиями к надежности. Периферийные методы (группа Б) отличаются более низкими капитальными затратами за счет переноса вычислений на IoT-шлюзы, однако распределенная инфраструктура может увеличить операционные расходы на обслуживание и обновление шлюзов. Методы на основе надстроечных СУБД-ВР (группа В) наиболее экономичны для модернизации существующих систем, так как

используют уже развернутую инфраструктуру реляционных баз данных. Гибридные методы (группа Г) обеспечивают баланс капитальных и операционных затрат: требуют умеренных вложений на старте, но в долгосрочной перспективе снижают риски потерь данных и эксплуатационные издержки. Таким образом, предложенная классификация может служить основой для технико-экономического обоснования проектов создания цифровых двойников, позволяя увязать архитектурные решения с затратами предприятия.

Проведенное исследование позволило систематизировать существующие подходы к интеграции данных в цифровых двойниках. Показано, что ключевыми методами являются потоковая передача данных, периферийная обработка и их гибридные комбинации. На основе анализа литературных источников предложена авторская

классификация, в которой методы интеграции различаются по способу передачи данных, типу хранения и топологии сети.

Предложенная классификация позволяет обоснованно выбирать архитектуру интеграции в зависимости от требований к латентности, масштабируемости и необходимости интеграции с существующими информационными системами. Новизна работы состоит в комплексном учете трех разнородных критериев в рамках единой классификации, а также в добавлении экономической оценки различных архитектурных решений. Практическая значимость заключается в возможности использования предложенной классификации при технико-экономическом обосновании проектов цифровых двойников, что позволяет оптимизировать капитальные и операционные затраты предприятий.

Библиографический список

1. Воробчикова Е. О. Персонализация как ключевой принцип обучения взрослых в условиях цифровизации // Вестник Мининского университета. — 2025. — Т. 13, № 1. — С. 1. — DOI: [10.26795/2307-1281-2025-13-1-1](#).
2. Дизайн цифрового двойника / С. Малакути [и др.] // Современная электроника. — 2022. — № 8. — С. 36–45.
3. Зорина Н. В., Забурдяев А. В. Информационно-аналитическая система организации конвейерной обработки данных для умного производства : сборник трудов РТУ МИРЭА // Цифровые технологии и системы. — М., 2023. — С. 320–342.
4. Иванов Е. В., Цымблер М. Л. Обзор современных систем обработки временных рядов // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. — 2020. — Т. 9, № 4. — С. 79–97. — DOI: [10.14529/cmse200406](#).
5. Лычкина Н. Н., Павлов В. В. Концепция цифрового двойника и роль имитационных моделей в архитектуре цифрового двойника // Прикладная информатика. — 2024. — № 1. — С. 4–23.
6. Мударисов А. А. Методы интеграции данных IoT в цифровые двойники для моделирования производственных процессов : сборник трудов XV Международной научно-практической конференции // Молодежь и современные информационные технологии. — Казань : Казанский государственный энергетический университет, 2024. — С. 108–111.
7. Паршина И. С., Фролов Е. Б. Разработка цифрового двойника производственной системы на базе современных цифровых технологий // Экономика в промышленности. — 2020. — Т. 13, № 1. — С. 29–34. — DOI: [10.17073/2072-1633-2020-1-29-34](#).
8. Петров А. В. Цифровые двойники — цифровые тени — цифровые двойники // Актуальные вопросы аграрной науки. — 2024. — 3(52). — С. 73–81. — DOI: [10.51215/2411-6483-2024-52-73-81](#).
9. Пономарев К. С., Феофанов А. Н., Гришина Т. Г. Стратегия цифрового двойника производства как метод цифровой трансформации предприятия // Вестник МГТУ «Станкин». — 2019. — 2(46). — С. 23–30.
10. Прохоров А., Лысачев М. Цифровой двойник. Анализ, тренды, мировой опыт / под ред. А. Боровкова. — М. : АльянсПринт, 2020. — 401 с. — ISBN 978-5-6042891-7-4.
11. Цифровые двойники и дискретно-событийные системы / Д. Е. Намиот [и др.] // International Journal of Open Information Technologies. — 2021. — Т. 9, № 2. — С. 70–75.