

УДК 004.032.26

5.2.2. Математические, статистические и инструментальные методы в экономике (физико-математические науки, экономические науки)

ПРИМЕНЕНИЕ MCP-СЕРВЕРА ДЛЯ ОРГАНИЗАЦИИ УПРАВЛЯЕМОГО ДОСТУПА ЯЗЫКОВОЙ МОДЕЛИ К КОНТЕКСТУ 1С

Попова Ольга Борисовна

Кандидат технических наук, доцент кафедры
информационных систем и программирования

РИНЦ-SCIENCE INDEX SPIN-код: 1095-6863

popova_ob@mail.ru

*ФГБОУ ВО «Кубанский государственный
технологический университет», Краснодар, Россия*

Янаева Марина Викторовна

Кандидат технических наук, доцент кафедры
информационных систем и программирования,
заведующий кафедрой

*ФГБОУ ВО «Кубанский государственный
технологический университет», Краснодар, Россия*

Волкодав Никита Владимирович

студент кафедры информационных систем и
программирования

nikitaphedorov02@gmail.com

*ФГБОУ ВО «Кубанский государственный
технологический университет», Краснодар, Россия*

Статья решает прикладной вопрос сопровождения 1С: как дать языковой модели сведения о конфигурации и не передать ей права на учетную базу. Промежуточным слоем выбран MCP-сервер. Он публикует для модели не общий доступ к данным, а набор инструментов: поиск метаданных, просмотр структуры объекта, поиск связей, чтение синтетических записей и подготовку dry-run запроса к Bitrix. Проверка проведена на локальном прототипе. В него входят слой метаданных 1С, синтетические записи, HTTP- и MCP-интерфейсы, а также тестовый контур Bitrix REST API. Прототип изолирован от производственных систем: работает с обезличенными данными, не выполняет произвольный SQL и пользовательский код, а запросы к внешним сервисам формирует без отправки

Ключевые слова: ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, ЯЗЫКОВАЯ МОДЕЛЬ, 1С, MCP-СЕРВЕР, МЕТАДААННЫЕ, BITRIX, REST API, ДЕМОСТРАЦИОННЫЙ ПРОТОТИП, БЕЗОПАСНЫЙ РЕЖИМ

<http://dx.doi.org/10.21515/1990-4665-221-021>

UDC 004.032.26

5.2.2. Mathematical, statistical and instrumental methods of economics (physical and mathematical sciences, economic sciences)

APPLICATION OF AN MCP SERVER FOR ORGANIZING CONTROLLED ACCESS OF A LANGUAGE MODEL TO THE 1C CONTEXT

Popova Olga Borisovna

Candidate of Technical Sciences, Associate Professor of
the Department of Information Systems and

Programming

RSCI-SCIENCE INDEX SPIN-code: 1095-6863

popova_ob@mail.ru

*Kuban State Technological University, Krasnodar,
Russia*

Yanaeva Marina Viktorovna

Candidate of Technical Sciences, Associate Professor of
the Department of Information Systems and
Programming, Head of the Department

*Kuban State Technological University, Krasnodar,
Russia*

Volkodav Nikita Vladimirovich

student of the Department of Information Systems and
Programming

nikitaphedorov02@gmail.com

*Kuban State Technological University, Krasnodar,
Russia*

The article addresses a practical issue in 1C support: how to provide a language model with configuration context without giving it rights to the accounting database. The intermediate layer is an MCP server. It exposes not general data access, but a set of tools: metadata search, object structure inspection, relationship analysis, synthetic record reading, and Bitrix dry-run request preparation. The scheme is tested on a local prototype. It includes a 1C metadata layer, synthetic records, HTTP and MCP interfaces, and a test Bitrix REST API contour. The prototype is isolated from production systems: it works with anonymized data, does not execute arbitrary SQL or user code, and prepares external service requests without sending them

Keywords: ARTIFICIAL INTELLIGENCE, LANGUAGE MODEL, 1C, MCP SERVER, METADATA, BITRIX, REST API, DEMONSTRATION PROTOTYPE, SAFE MODE

Введение

В сопровождении 1С-конфигураций запрос редко заканчивается простым поиском документа или справочника. Специалист восстанавливает цепочку: какие реквизиты участвуют в операции, куда ведет ссылка на контрагента или договор, какие движения попали в регистры, с каким обращением пользователя связана правка. В развитой базе именно поиск такого контекста часто занимает больше времени, чем последующее действие в системе.

На платформе «1С:Предприятие» строятся прикладные решения для учета, продаж, закупок, складских операций, производства, кадрового и финансового контура [1]. После доработок под конкретную организацию структура базы обычно отходит от типовой. Один документ связан с организацией, контрагентом, договором, складом, номенклатурой, регистрами и дополнительными объектами. Поэтому в сопровождении нужен не только поиск по имени, но и быстрый просмотр состава объекта и его зависимостей.

Языковая модель полезна там, где пользователь описывает задачу обычным языком, а система должна найти технический контекст: объект метаданных, связанные реквизиты, пояснение для обращения или черновик инструкции. Работы по retrieval-augmented generation и корпоративным LLM-сценариям показывают, что внешний источник знаний связывает ответ с данными организации и уменьшает риск вымышленных сведений [7-9]. При этом учетную базу нельзя открывать модели напрямую. В ней хранятся финансовые сведения, персональные данные, документы и регистры; неверно понятый запрос способен повредить данные или раскрыть больше информации, чем требуется для ответа.

Поэтому доступ модели к 1С целесообразно оформить через промежуточный слой, где заранее перечислены допустимые операции. В

данной работе такую роль выполняет MCP-сервер. Model Context Protocol задает механизм, при котором клиент получает список инструментов и вызывает выбранный инструмент с параметрами [2]. В контуре 1С это отделяет языковую модель от производственной базы: проверка параметров, выбор источника и сбор ответа остаются на стороне сервера.

Цель и задачи исследования

Цель работы - разработать и проверить на демонстрационном прототипе схему, при которой языковая модель обращается к контексту 1С через MCP-сервер, а не через прямое подключение к учетной базе.

Для этого в работе решаются пять задач:

1. Выделить риски прямого подключения языковой модели к корпоративному контексту 1С.
2. Описать архитектуру MCP-сервера со слоем метаданных, демонстрационными записями и тестовым контуром Bitrix.
3. Подобрать набор инструментов для поиска метаданных, просмотра структуры объектов, анализа связей и dry-run интеграционной операции.
4. Описать ограничения безопасного режима: запрет произвольного SQL, выполнения кода и обращения к производственным системам.
5. Определить область применения прототипа при сопровождении 1С-конфигураций и направления дальнейшей разработки.

Методология

В архитектуре прототипа MCP-сервер расположен между пользователем, языковой моделью и источниками корпоративного контекста. Модель не подключается напрямую к базе 1С, SQL-хранилищу или CRM-сервису. Она выбирает инструмент из опубликованного списка; сервер проверяет параметры, обращается к разрешенному источнику и возвращает результат. По этой логике прототип близок к tool-augmented

LLM: модель использует внешний инструмент или API как контролируемую операцию, а не как свободный канал доступа [10, 11]. Роль модели ограничена интерфейсом к описанным функциям.

Проверка выполнена на локальном прототипе. В его состав включены слой метаданных, слой демонстрационных записей, HTTP-интерфейс, MCP-инструменты, тестовый контур Bitrix REST API и безопасный режим запуска. Слой метаданных хранит сведения об объектах 1С: документах, справочниках, регистрах, реквизитах, SQL-именах и связях между сущностями. Поиск работает по техническому имени, синониму, назначению, описанию и связанным реквизитам.



Рисунок 1 – Общая архитектура демонстрационного прототипа

Слой демонстрационных записей содержит синтетический фрагмент учетной системы: организации, контрагентов, склады, номенклатуру, заказы, счета, реализации и оплаты. Эти записи не связаны с реальной информационной базой. На них показываются сценарии, где пользователь работает не только с описанием объекта, но и с условной карточкой или документом.

Тестовый контур Bitrix нужен как пример внешней CRM-интеграции. В Bitrix24 REST API предусмотрены методы работы с CRM-элементами: например, `crm.item.update` обновляет элемент по идентификатору типа сущности, идентификатору элемента и набору изменяемых полей [3]. В прототипе такой вызов не отправляется во внешний сервис. Сервер собирает предварительный запрос, чтобы показать состав будущей интеграционной операции до ее выполнения.

Безопасный режим работы

В прототипе безопасный режим означает изоляцию от производственной инфраструктуры. Сервер запускается локально, использует синтетические данные и не требует подключения к рабочей базе 1С, реальному Bitrix-порталу, внешним токенам или закрытым сетевым ресурсам.

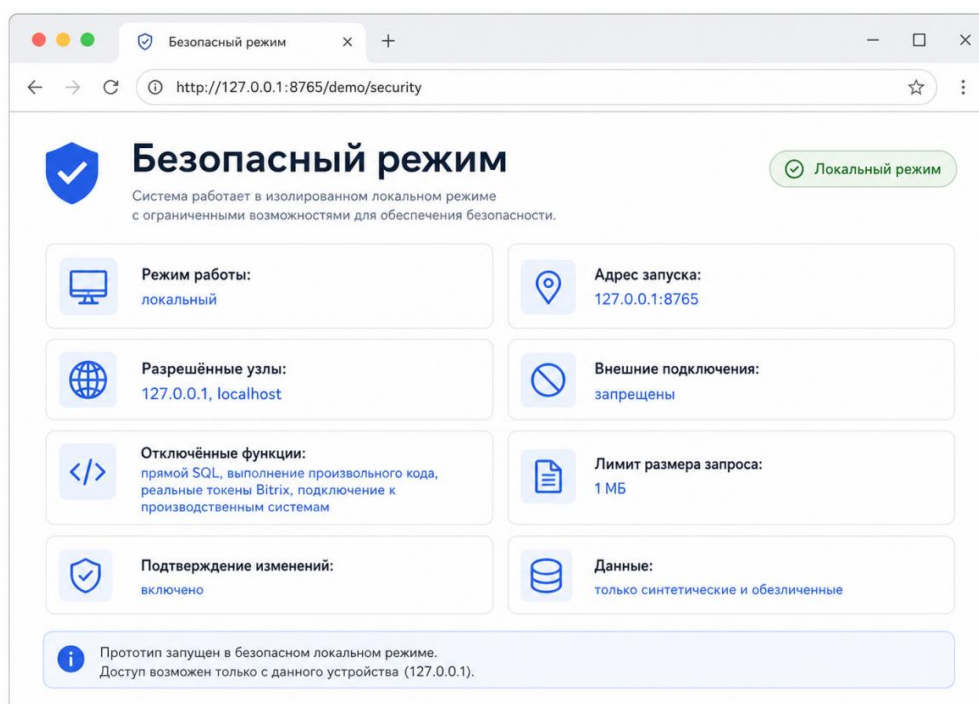


Рисунок 2 – Проверка безопасного режима запуска прототипа

Сервер принимает запросы только на локальном адресе. Такая настройка снижает вероятность случайной публикации приложения во внешнюю сеть. Дополнительно проверяется заголовок Host,

ограничивается размер входящего запроса и задается базовый набор HTTP-заголовков безопасности. Это не промышленная защита, а минимальная граница демонстрационного контура: без нее учебный прототип легко становится рискованным интеграционным экспериментом.

Из демонстрационного режима исключены операции, способные затронуть реальные данные: произвольный SQL, выполнение кода, подключение к рабочей базе 1С, вызов внешних REST-методов и изменение неразрешенных полей. Изменения допускаются только в синтетических записях либо в формате предварительного просмотра будущего запроса. Поэтому прототип можно показывать и тестировать без воздействия на производственную среду.

Результаты

В результате получен прототип, где контекст 1С доступен через ограниченный набор MCR-инструментов. Для проверки принципа выбраны функции, закрывающие базовый цикл сопровождения: проверка доступности сервера, поиск метаданных, просмотр структуры объектов, поиск связей, чтение демонстрационных записей и подготовка Bitrix-запроса без отправки. Состав инструментов приведен в таблице 1.

Таблица 1 – Основные MCR-инструменты демонстрационного прототипа

Инструмент	Назначение	Тип операции
ping	Проверяет, что сервер запущен, и возвращает краткое состояние прототипа.	Чтение
list_objects	Возвращает список объектов метаданных с фильтрацией по типу.	Чтение
search_metadata	Ищет объекты по названию, синониму, описанию, SQL-имени, тегам и реквизитам.	Поиск
get_object_structure	Показывает структуру выбранного объекта: реквизиты, группы полей и ссылки.	Анализ
find_related	Находит входящие и исходящие связи между объектами.	Анализ
list_records, get_record	Выводит список синтетических записей и карточку выбранного экземпляра.	Чтение
bitrix_dry_run_update	Собирает предварительный запрос к Bitrix без отправки во внешний сервис.	Предпросмотр

Набор из таблицы 1 специально не сделан универсальным интерфейсом к учетной системе. В нем нет функции для произвольного SQL-запроса, запуска кода, вызова любого метода Bitrix или изменения произвольного объекта 1С. Каждый инструмент решает одну прикладную задачу и принимает заранее определенные параметры.

Например, `search_metadata` принимает текстовый запрос и возвращает найденные объекты с объяснением совпадений. Для запроса по заказу покупателя инструмент выводит соответствующий документ, его синоним, описание и связанные реквизиты. `get_object_structure` показывает логическое имя, тип объекта, SQL-имя, реквизиты и ссылки на другие сущности.

В сопровождении 1С-конфигураций такой результат закрывает рутинную, но трудоемкую часть работы: восстановление контекста. Специалисту нужно понять, где используется объект, какие поля влияют на процесс и какие документы или регистры связаны между собой. Языковая модель формулирует ответ для пользователя, а чтение данных и подготовка действий остаются на стороне контролируемого программного слоя.

Обсуждение

Сравнение с аналогами

Схему МСР-сервера удобно сопоставить с другими способами передать модели корпоративный контекст. В научных работах близкая проблематика идет по двум линиям: подключение внешней памяти и корпоративных хранилищ к генеративной модели [7-9], а также выбор и вызов инструментов или API самой моделью [10-13]. В прикладной интеграции первым кандидатом обычно становится прямой REST API внешней системы. Он удобен, когда методы описаны и права доступа настроены. Недостаток в том, что ошибка интерпретации

пользовательского запроса сразу приближает модель к операции изменения данных. В прототипе REST-вызов вынесен в отдельный инструмент и представлен как `dry run` - предварительный просмотр без отправки.

Второй вариант - ODBC, прямой SQL-доступ или другое низкоуровневое подключение к хранилищу. В отчетности и техническом анализе такой канал оправдан, но для языковой модели он опасен. Запрос к физической структуре базы требует знания таблиц, прав, ограничений целостности и особенностей конкретной конфигурации. В разработанной архитектуре SQL-имена присутствуют только в описании метаданных; инструмента для выполнения произвольного SQL нет.

Третий путь - текстовый экспорт из 1С. Описания объектов, инструкции, фрагменты карточек и обращения пользователей выгружаются в файл или поисковый индекс. Риск прямого воздействия на учетную систему здесь ниже, зато появляются две проблемы: экспорт устаревает и теряет часть связей между объектами. МСР-сервер оставляет обращение к актуализируемому контексту, но каждое действие привязывает к конкретному инструменту.

Близкая альтернатива - пользовательские HTTP-сервисы 1С и интеграционные адаптеры. Они также выносят бизнес-логику в отдельный программный слой. Разница в форме контракта: в МСР-схеме инструменты описаны так, чтобы языковая модель могла выбрать нужную операцию, но не получила универсальный интерфейс к учетной системе. Поисковые, аналитические и демонстрационные операции разделены заранее.

МСР-сервер в этой работе не заменяет штатные механизмы интеграции 1С, Bitrix или внешних сервисов. Его роль уже: поставить управляемую оболочку между языковой моделью и корпоративным контекстом. Модель формулирует запрос и интерпретирует ответ, но не

получает самостоятельного доступа к данным и операциям изменения.

Следующий шаг для поискового слоя - векторный индекс. Например, Qdrant хранит представления объектов и ищет их по смысловой близости [4]. Для 1С это особенно полезно, когда пользователь не знает точного названия регистра, документа или справочника и описывает задачу обычным языком. Эта логика согласуется с RAG-архитектурами, где ответ строится с учетом извлеченного внешнего контекста [7-9]. Найденный фрагмент, однако, не должен запускать изменение данных: запись в 1С или Bitrix допустима только через проверенный инструмент.

Практическая значимость и дальнейшее применение

Практическая ценность схемы связана с разделением ролей. Языковая модель работает как интерфейс к техническому контексту 1С, а не как участник учетной операции. Пользователь задает вопрос в свободной форме, сервер переводит его в вызов ограниченного инструмента, а результат возвращается в виде, пригодном для пояснения или проверки специалистом. За счет этого сокращается время поиска метаданных, просмотра структуры документов и справочников, анализа связей и подготовки первичных ответов на обращения.

Дальнейшая разработка прототипа предполагает подключение тестовой базы 1С через контролируемый HTTP-интерфейс, использование входящих вебхуков Bitrix24 [5], журналирование действий, подтверждение операций и защиту от повторного применения одного запроса. При расширении сохраняется базовое ограничение: модель предлагает действие, а запись выполняет только проверенный инструмент с заданными параметрами.

Заключение

Разработанный вариант не рассчитан на промышленное внедрение без доработки. В статье показан прикладной принцип: языковая модель используется в сопровождении 1С-конфигураций, но не получает прямой доступ к производственным системам. Локальный запуск, синтетические данные, ограниченные инструменты и предварительный просмотр внешних запросов задают безопасную основу для дальнейшей разработки выпускной квалификационной работы.

Поставленные задачи решены: выделены риски прямого доступа языковой модели к контексту 1С, описана архитектура MCP-сервера, определен набор демонстрационных инструментов, зафиксированы ограничения безопасного режима и обозначены направления дальнейшего применения прототипа.

Исследование выполнено при финансовой поддержке Кубанского научного фонда в рамках научного проекта № Н-25.1/19.

Список литературы

1. 1С:Enterprise Application Platform for Business Automation Software Creation. – URL: <https://www.1ci.com/developers/developers/> (дата обращения: 01.06.2026). – Текст : электронный.
2. Model Context Protocol. Server tools specification. – URL: <https://modelcontextprotocol.io/specification/2025-06-18/server/tools> (дата обращения: 03.06.2026). – Текст : электронный.
3. Bitrix24 REST API. Update CRM Item. – URL: <https://apidocs.bitrix24.com/api-reference/crm/universal/crm-item-update.html> (дата обращения: 04.06.2026). – Текст : электронный.
4. Qdrant Documentation. Vector Search. – URL: <https://qdrant.tech/documentation/overview/vector-search/> (дата обращения: 05.06.2026). – Текст : электронный.
5. Bitrix24 REST API. Incoming and Outgoing Webhooks. – URL: <https://apidocs.bitrix24.com/local-integrations/local-webhooks.html> (дата обращения: 07.06.2026). – Текст : электронный.
6. FastAPI Documentation. Testing. – URL: <https://fastapi.tiangolo.com/tutorial/testing/> (дата обращения: 07.06.2026). – Текст : электронный.
7. Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Kuttler H., Lewis M., Yih W., Rocktaschel T., Riedel S., Kiela D. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks // Advances in Neural Information Processing Systems. - 2020. - Vol. 33. - P. 9459-9474. - Текст : непосредственный.
8. Klesel M., Wittmann H. F. Retrieval-Augmented Generation (RAG) // Business &

Information Systems Engineering. - 2025. - Vol. 67. - P. 551-561. - DOI: 10.1007/s12599-025-00945-3. - Текст : непосредственный.

9. Karakurt E., Akbulut A. Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs) for Enterprise Knowledge Management and Document Automation: A Systematic Literature Review // Applied Sciences. - 2026. - Vol. 16, No. 1. - Article 368. - DOI: 10.3390/app16010368. - Текст : электронный.

10. Schick T., Dwivedi-Yu J., Dessi R., Raileanu R., Lomeli M., Hambro E., Zettlemoyer L., Cancedda N., Scialom T. Toolformer: Language Models Can Teach Themselves to Use Tools // Advances in Neural Information Processing Systems. - 2023. - Vol. 36. - URL: <https://arxiv.org/abs/2302.04761> (дата обращения: 22.06.2026). - Текст : электронный.

11. Yao S., Zhao J., Yu D., Du N., Shafran I., Narasimhan K., Cao Y. ReAct: Synergizing Reasoning and Acting in Language Models // Proceedings of the 11th International Conference on Learning Representations (ICLR 2023). - URL: https://openreview.net/forum?id=WE_vluYUL-X (дата обращения: 22.06.2026). - Текст : электронный.

12. Qin Y., Liang S., Ye Y., Zhu K., Yan L., Lu Y., Lin Y., Cong X., Tang X., Qian B., Zhao S., Hong L., Tian R., Xie R., Zhou J., Gerstein M., Li D., Liu Z., Sun M. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs // Proceedings of the International Conference on Learning Representations (ICLR 2024). - URL: <https://openreview.net/forum?id=dHng2O0Jjr> (дата обращения: 22.06.2026). - Текст : электронный.

13. Patil S. G., Zhang T., Wang X., Gonzalez J. E. Gorilla: Large Language Model Connected with Massive APIs // Advances in Neural Information Processing Systems. - 2024. - URL: <https://openreview.net/forum?id=tBRNC6YemY> (дата обращения: 22.06.2026). - Текст : электронный.

List of literature

1. 1C:Enterprise Application Platform for Business Automation Software Creation. – URL: <https://www.1ci.com/developers/developers/> (accessed: 01.06.2026).

2. Model Context Protocol. Server tools specification. – URL: <https://modelcontextprotocol.io/specification/2025-06-18/server/tools> (accessed: 02.06.2026).

3. Bitrix24 REST API. Update CRM Item. – URL: <https://apidocs.bitrix24.com/api-reference/crm/universal/crm-item-update.html> (accessed: 04.06.2026).

4. Qdrant Documentation. Vector Search. – URL: <https://qdrant.tech/documentation/overview/vector-search/> (accessed: 05.06.2026).

5. Bitrix24 REST API. Incoming and Outgoing Webhooks. – URL: <https://apidocs.bitrix24.com/local-integrations/local-webhooks.html> (accessed: 07.06.2026).

6. FastAPI Documentation. Testing. – URL: <https://fastapi.tiangolo.com/tutorial/testing/> (accessed: 07.06.2026).

7. Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Kuttler H., Lewis M., Yih W., Rocktaschel T., Riedel S., Kiela D. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems. 2020. Vol. 33. P. 9459-9474.

8. Klesel M., Wittmann H. F. Retrieval-Augmented Generation (RAG). Business & Information Systems Engineering. 2025. Vol. 67. P. 551-561. DOI: 10.1007/s12599-025-00945-3.

9. Karakurt E., Akbulut A. Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs) for Enterprise Knowledge Management and Document Automation: A Systematic Literature Review. Applied Sciences. 2026. Vol. 16, No. 1. Article

368. DOI: 10.3390/app16010368.

10. Schick T., Dwivedi-Yu J., Dessi R., Raileanu R., Lomeli M., Hambro E., Zettlemoyer L., Cancedda N., Scialom T. Toolformer: Language Models Can Teach Themselves to Use Tools. *Advances in Neural Information Processing Systems*. 2023. Vol. 36. URL: <https://arxiv.org/abs/2302.04761> (accessed: 22.06.2026).

11. Yao S., Zhao J., Yu D., Du N., Shafran I., Narasimhan K., Cao Y. ReAct: Synergizing Reasoning and Acting in Language Models. *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*. URL: https://openreview.net/forum?id=WE_vluYUL-X (accessed: 22.06.2026).

12. Qin Y., Liang S., Ye Y., Zhu K., Yan L., Lu Y., Lin Y., Cong X., Tang X., Qian B., Zhao S., Hong L., Tian R., Xie R., Zhou J., Gerstein M., Li D., Liu Z., Sun M. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. *Proceedings of the International Conference on Learning Representations (ICLR 2024)*. URL: <https://openreview.net/forum?id=dHng2O0Jjr> (accessed: 22.06.2026).

13. Patil S. G., Zhang T., Wang X., Gonzalez J. E. Gorilla: Large Language Model Connected with Massive APIs. *Advances in Neural Information Processing Systems*. 2024. URL: <https://openreview.net/forum?id=tBRNC6YemY> (accessed: 22.06.2026).