Webinar #4

Implementing Generic APIs with Generative AI for Seamless Interoperability

Dr. Rodrigo Falcão, Fraunhofer IESE
November 28, 2024

Fraunhofer
IESE

Fraunhofer Institute for Experimental
Software Engineering IESE

About me…

- **Dr.-Ing. Rodrigo Falcão, PMP**

- Researcher and Project Manager at Fraunhofer IESE, Kaiserslautern, Germany

- ~15 years of industry experience prior to stepping into research

- Software architecture

- Lecturer of Software Architecture at Mannheim University of Applied Sciences, Germany

- **Lead researcher for "Generative AI in Software Architecture"**

Agenda

- **Interoperability 101**

- **An example scenario**

- **The problem**

- **What if…**

- **Generative AI**

- **Experiences**

- **Consequences**

- **What's next?**

Interoperability 101

Created with FhGenie / Dall-E

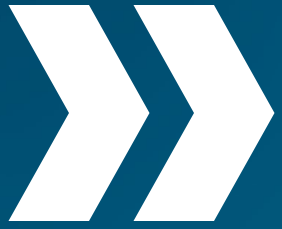# Quality is a core concern for stakeholders in a software project

Stakeholders' concerns

Functionality

- "WHAT"
- Key features of my system
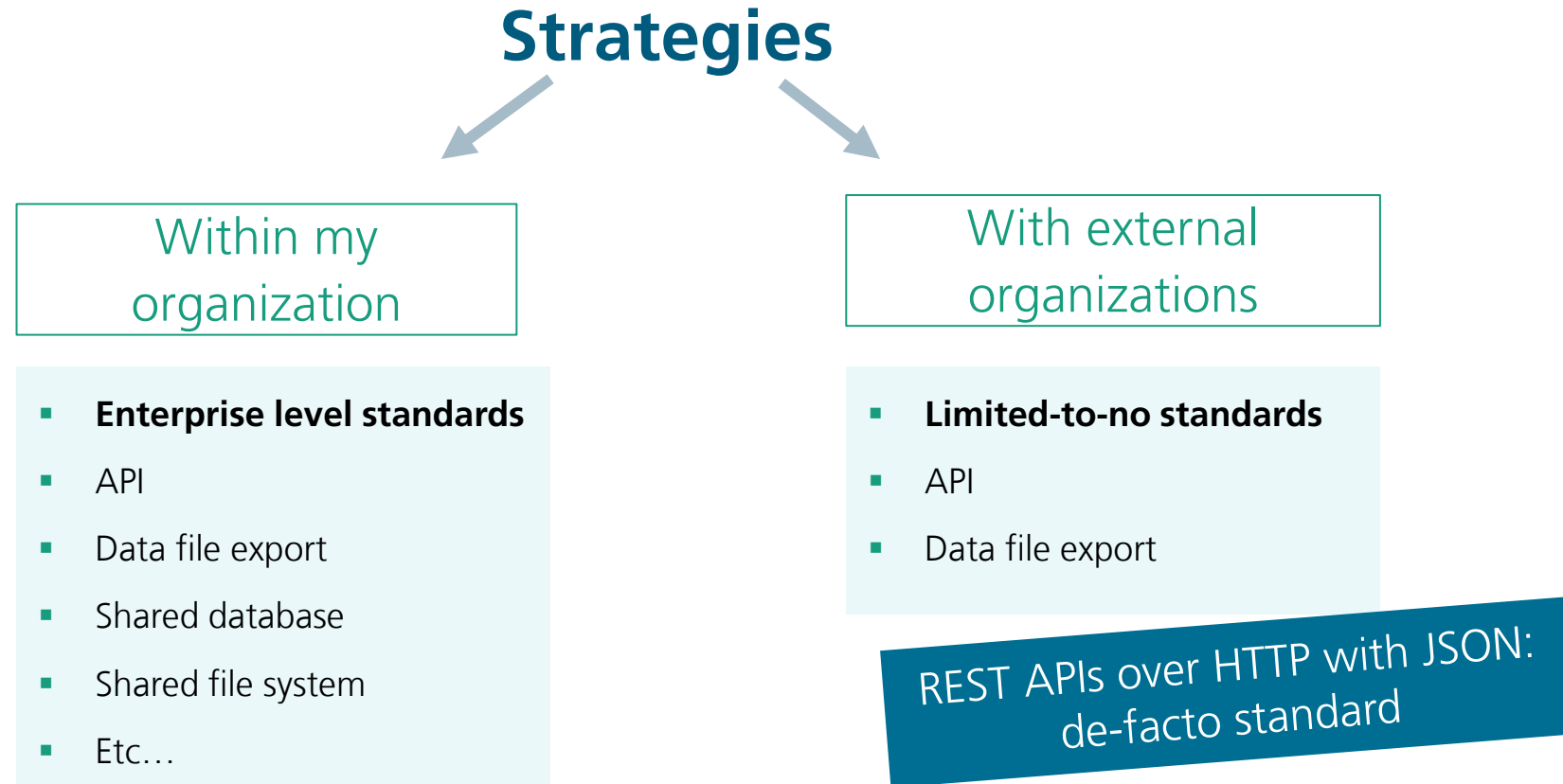- Makes my system unique

Quality

- "HOW GOOD"
- Maintainability
- Reliability
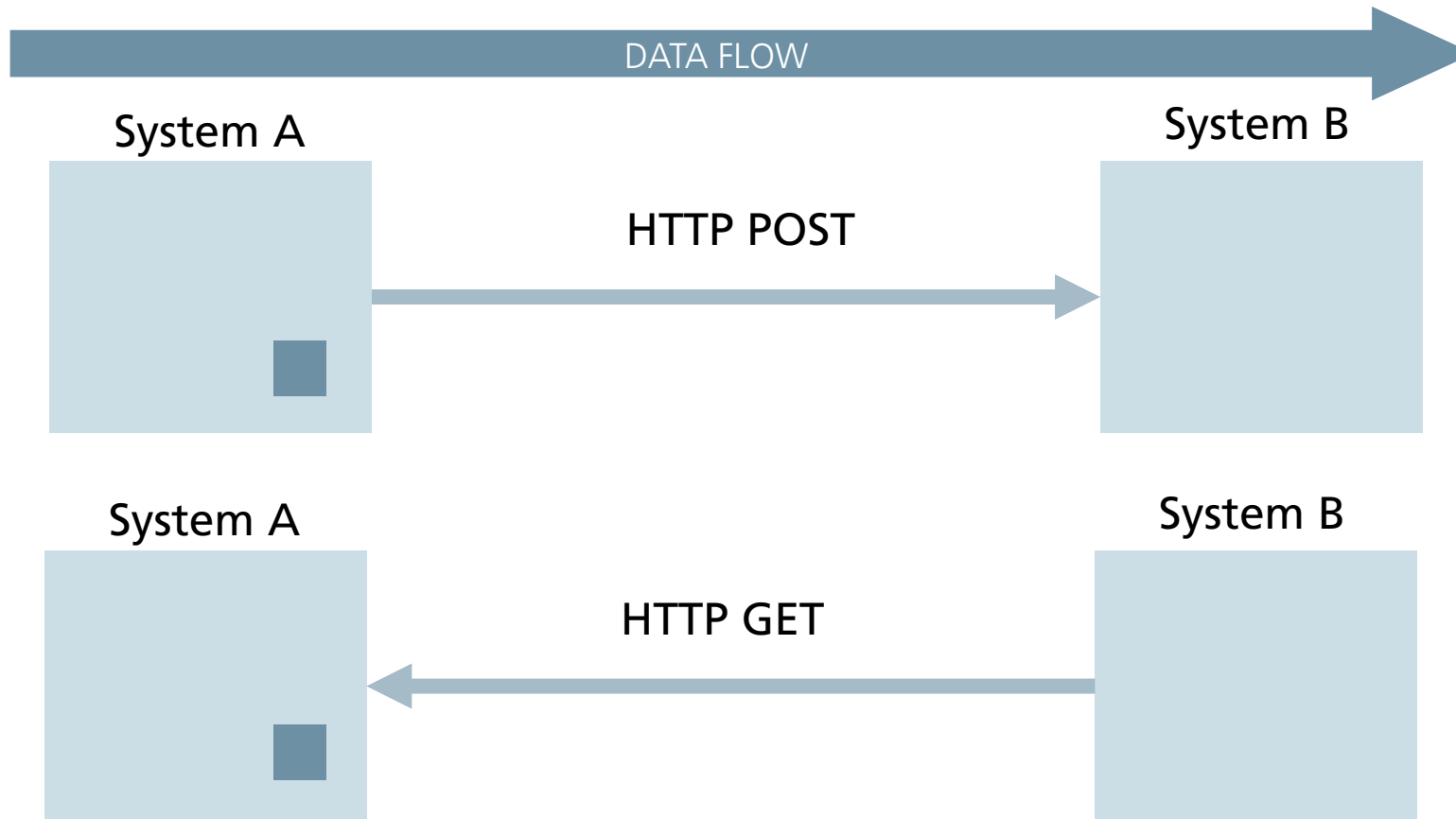- Correctness
- …
- **Interoperability**

Fraunhofer

capability of a product to exchange information with other products and mutually use the information that has been exchanged.

-- ISO/IEC 25010:2023

Public

Fraunhofer

APIs are among the most traditional strategies for data exchange with external organizations

Strategies

Within my organization

- **Enterprise level standards**
- API
- Data file export
- Shared database
- Shared file system
- Etc…

With external organizations

- **Limited-to-no standards**
- API
- Data file export

REST APIs over HTTP with JSON: de-facto standard

28.11.2024 © Fraunhofer Public

When a system needs another system's data, it can either expose or consume an API

DATA FLOW

System A

System B

HTTP POST

System A

System B

HTTP GET

Other designs are possible, but you got the idea

An example

(Agricultural Domain)

28.11.2024 © Fraunhofer IESE

Agriculture is a huge domain with several interoperability challenges

Agricultural domain

Huge

- Several subdomains
- Several processes
- Several formats
- Projects are silos
- …

Interoperability challenges

- Lack of standards
- Organizational challenges
- Technology heterogeneity
- Data quality

Fraunhofer
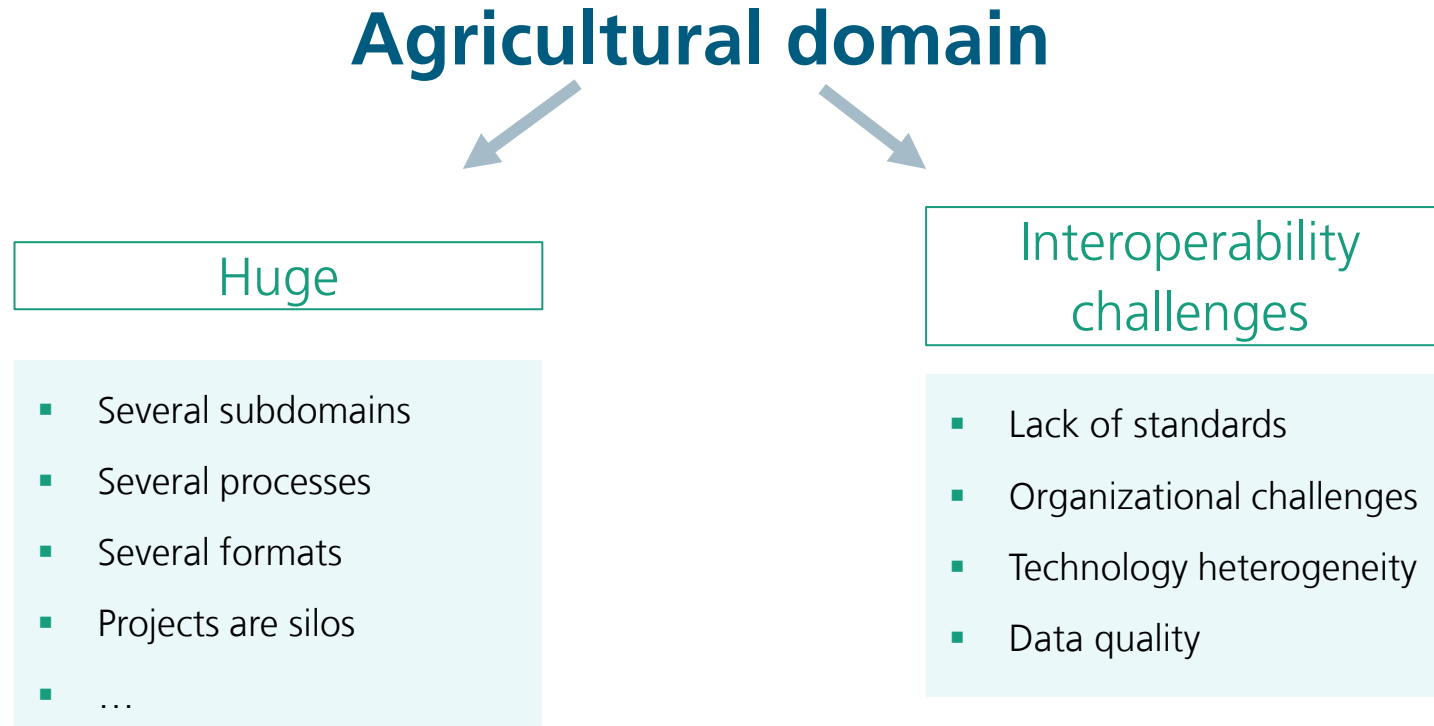
**Digital ecosystem
of Provider A**

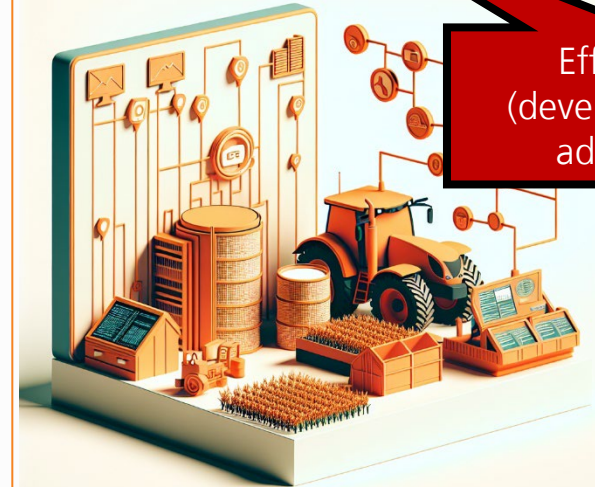**Digital ecosystem
of Provider B**

Effort-intensive
(development of data
adapters / APIs)

Effort-intensive
(development of data
adapters / APIs)
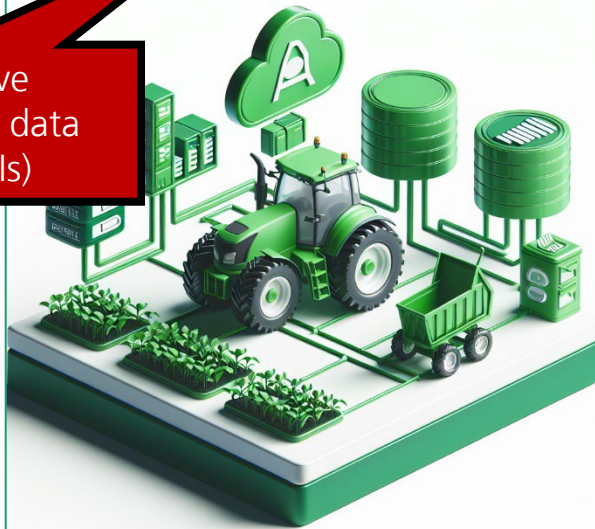
System A:

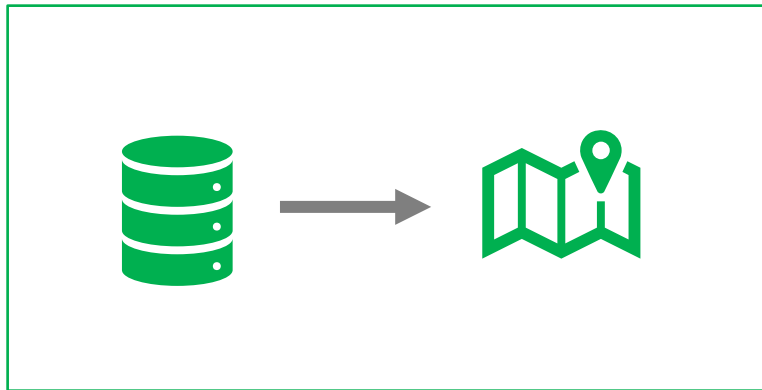Own standards, data
formats,schemas, semantics…

System B:

Own standards, data formats,
schemas, semantics…

Effort-intensive (manual)
data harmonization

Fraunhofer

Two solution providers implement Farm Management Information Systems to support farming activities
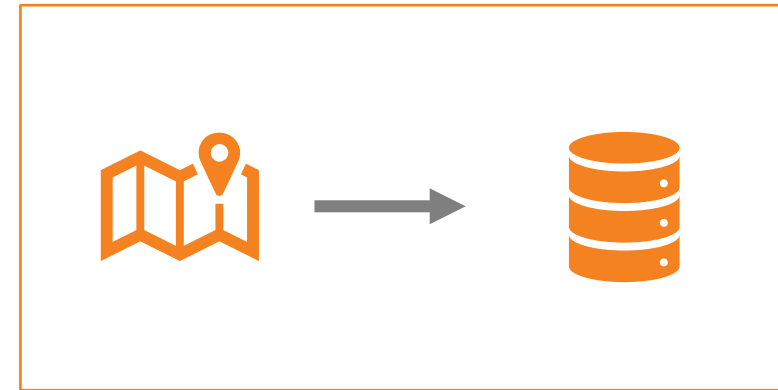
FMIS 1
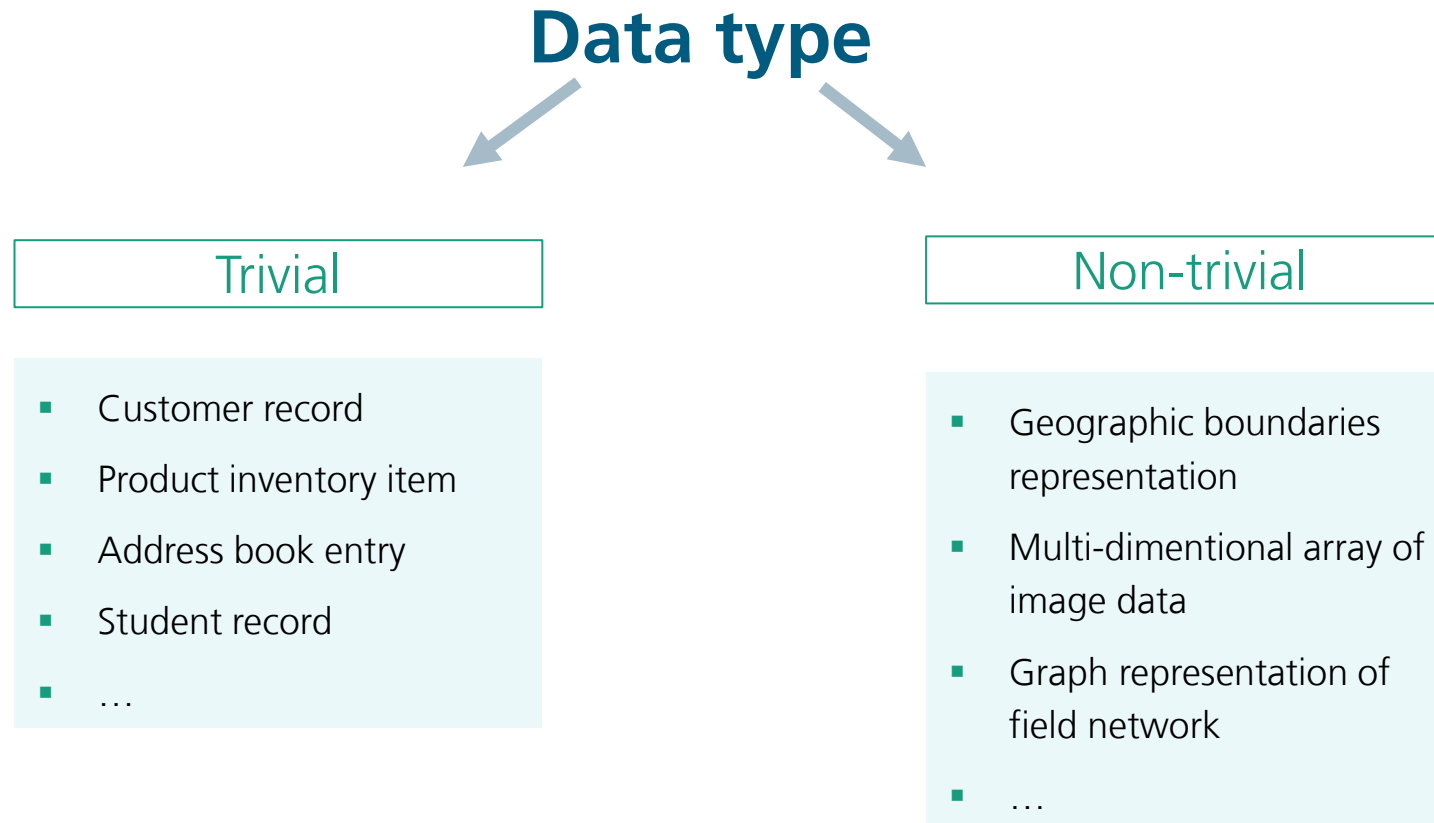
FMIS 2
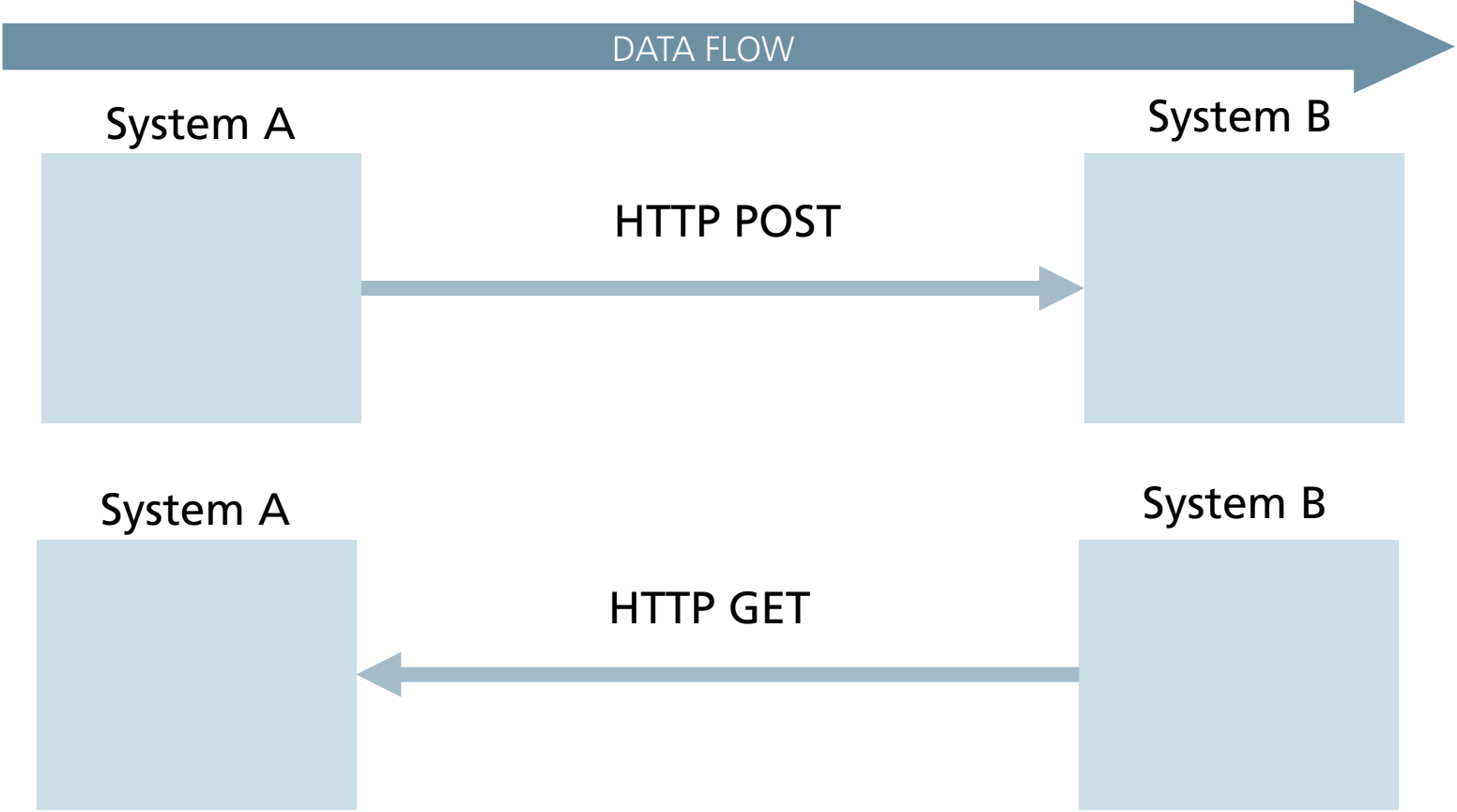
?

has field boundaries using its own schema

has its own representation of field boundaries

28.11.2024   © Fraunhofer   **Public**

Fraunhofer

Field boundaries are not the most trivial data type

Data type

Trivial

- Customer record
- Product inventory item
- Address book entry
- Student record
- …

Non-trivial

- Geographic boundaries representation
- Multi-dimentional array of image data
- Graph representation of field network
- …

The interface can be designed in different ways

DATA FLOW

System A

System B

HTTP POST

System A

System B

HTTP GET

28.11.2024   © Fraunhofer   **Public**

Consider that system B expose an API to receive field data

DATA FLOW

System A

System B

HTTP POST

Schema B'

Schema A

Schema B

28.11.2024   © Fraunhofer   **Public**   Fraunhofer

The problem

Created with FhGenie / Dall-E
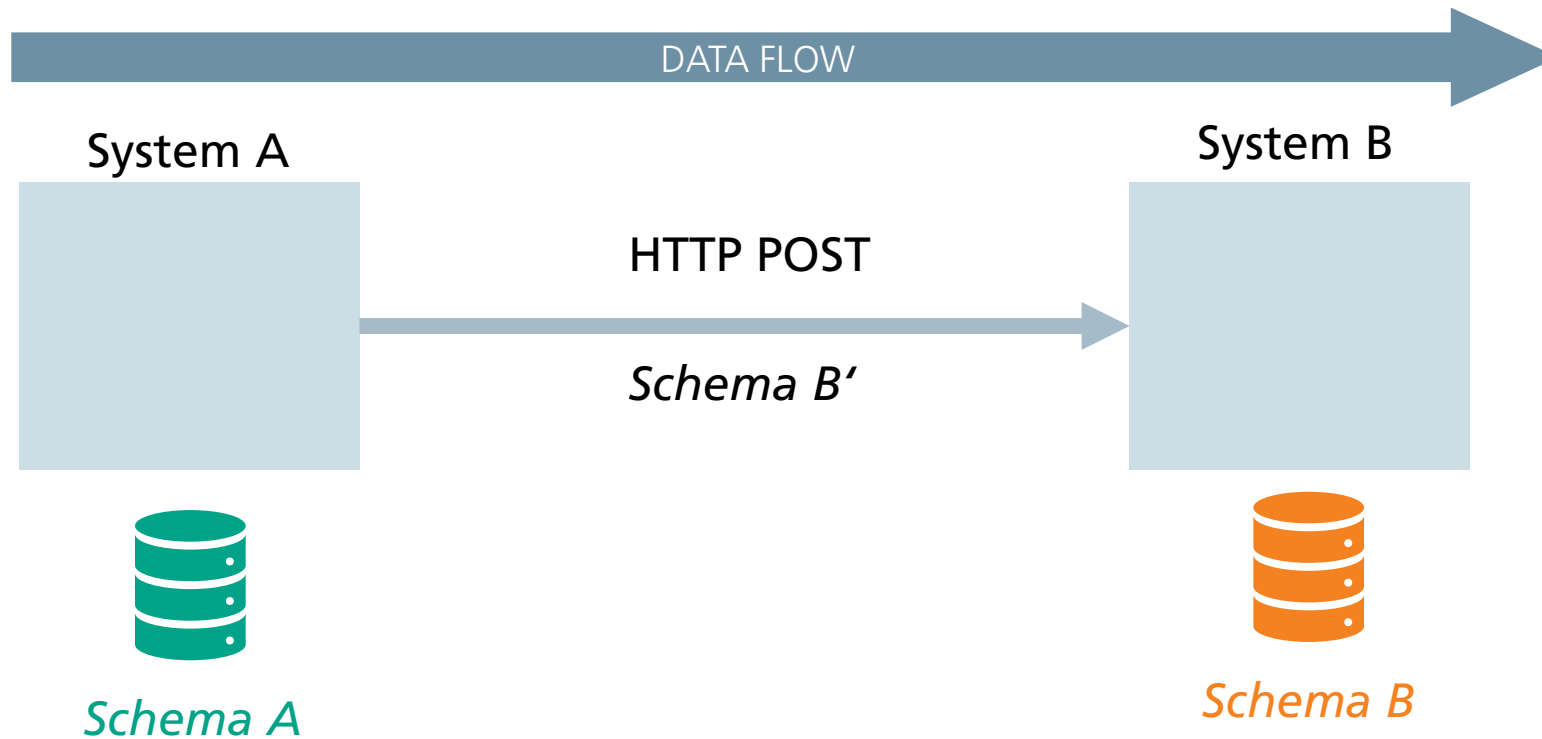
Achieving interoperability via APIs comes at a cost: implementation efforts

28.11.2024 © Fraunhofer **Public**

API consumers must invest efforts in complying with the API requirements

Providers

Consumers

Efforts

Functional aspects

- Define format
- Define schema

- Comply with requirements
- Convert data

Quality aspects

- take care of reliability

- must consider latency, fault tolerance, etc.

28.11.2024   © Fraunhofer   **Public**

Fraunhofer
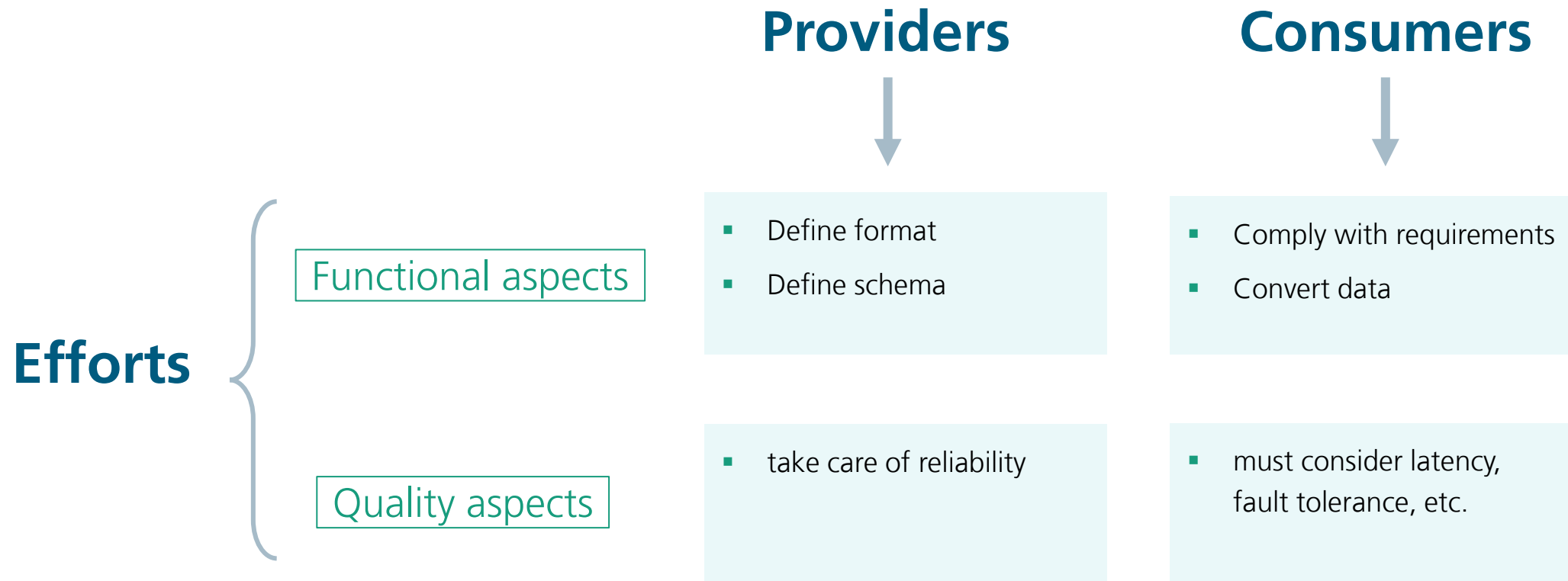
Who is willing to take the effort?

DATA FLOW

System A

System B

HTTP POST

Schema B'

Schema A

Schema B
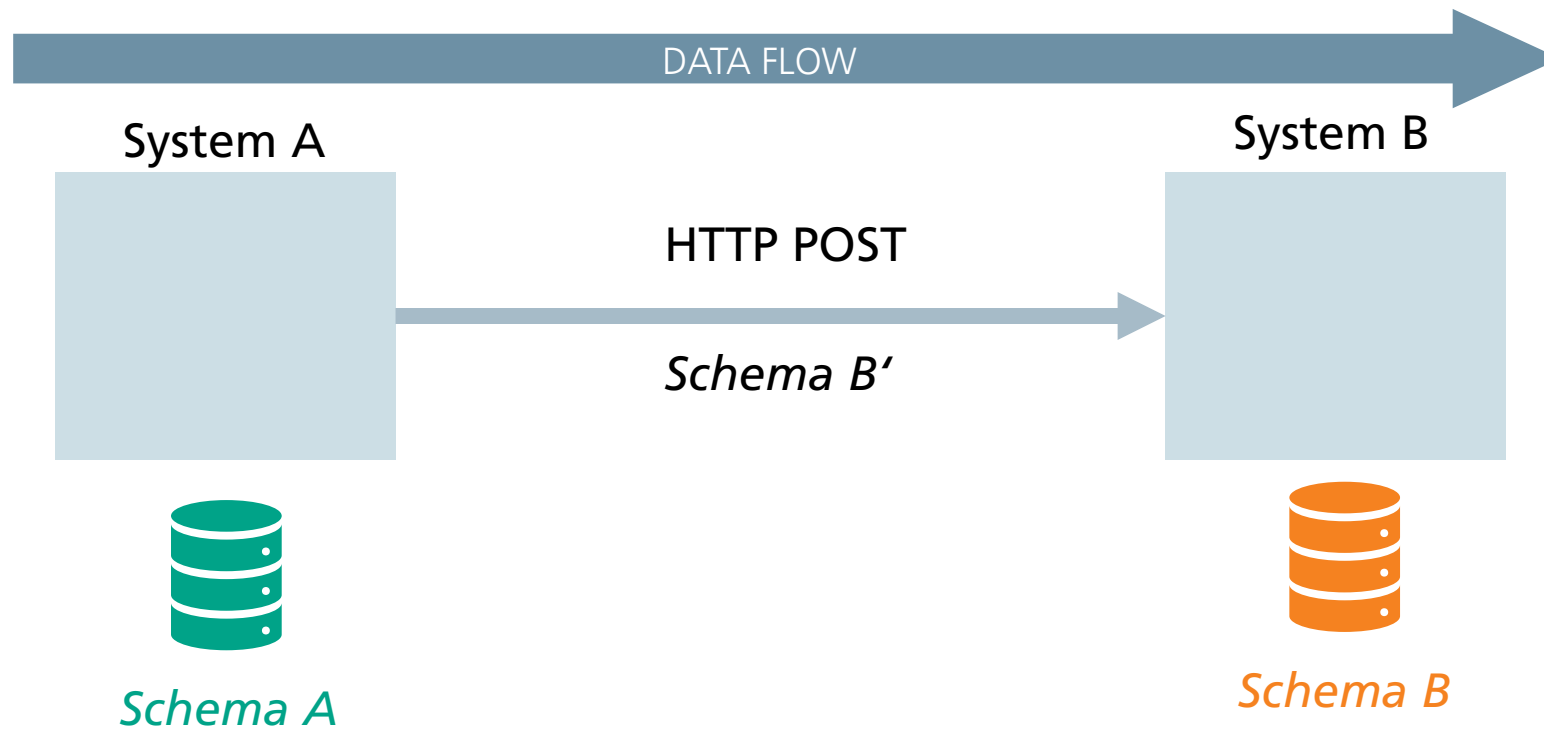
Typically, it is better to expose the API.

Fraunhofer

If I want field data, the dream is that I expose my API and everyone complies with it!

(But this is everyone's else dream!)

In the aftermath, it's not about exposing or consuming APIs; it's about the effort needed to understand a foreign schema and implement adapters for data conversion.
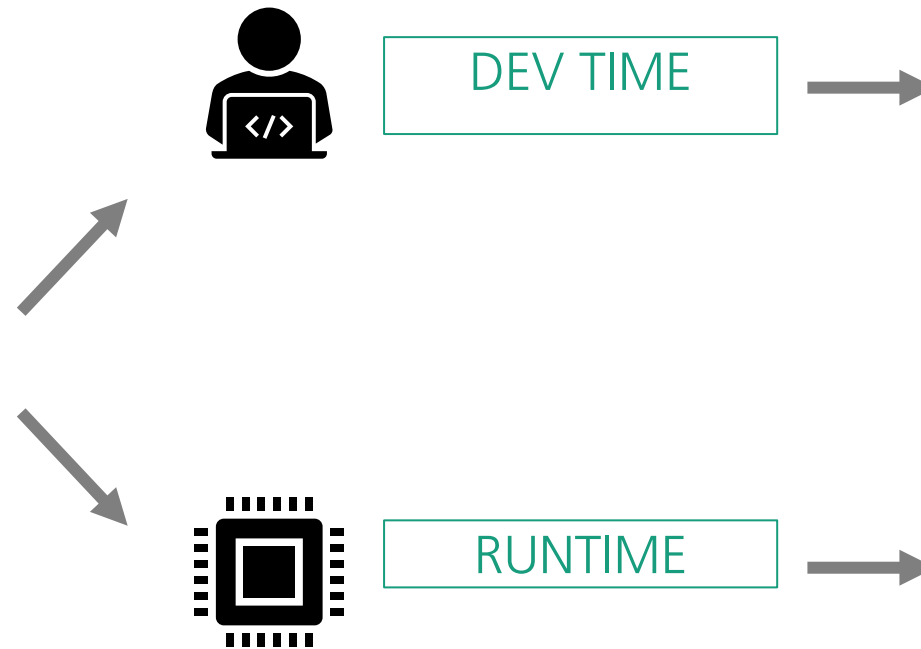
28.11.2024   © Fraunhofer

Public

Fraunhofer

What if…

Created with FhGenie / Dall-E

WHAT IF…

…our systems were smart to the point that new participants could join the agricultural ecosystem without development efforts to adapt interfaces?

# Recently, we at IESE have explored achieving interoperability at runtime

Interoperability

DEV TIME →

- Human effort to understand the interfaces, implement adequate data converters (a.k.a. "adapters"), test, deploy, etc.
- Deterministic solution
- Time to market ☹

RUNTIME →

- What if AI can do at runtime what humans would have to do at dev time?
- Can LLMs implement effective adapters at runtime?
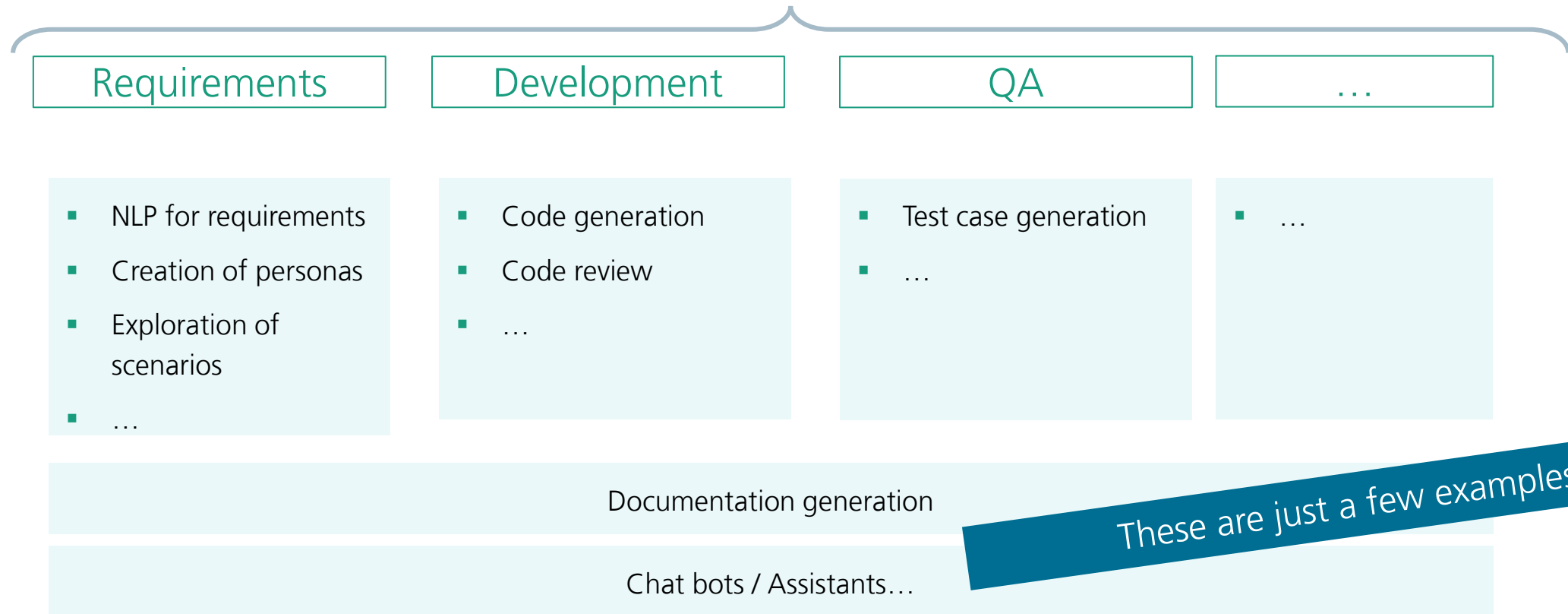
We have tested this idea in the agricultural domain

Public

Generative AI

Created with FhGenie / Dall-E

Fraunhofer
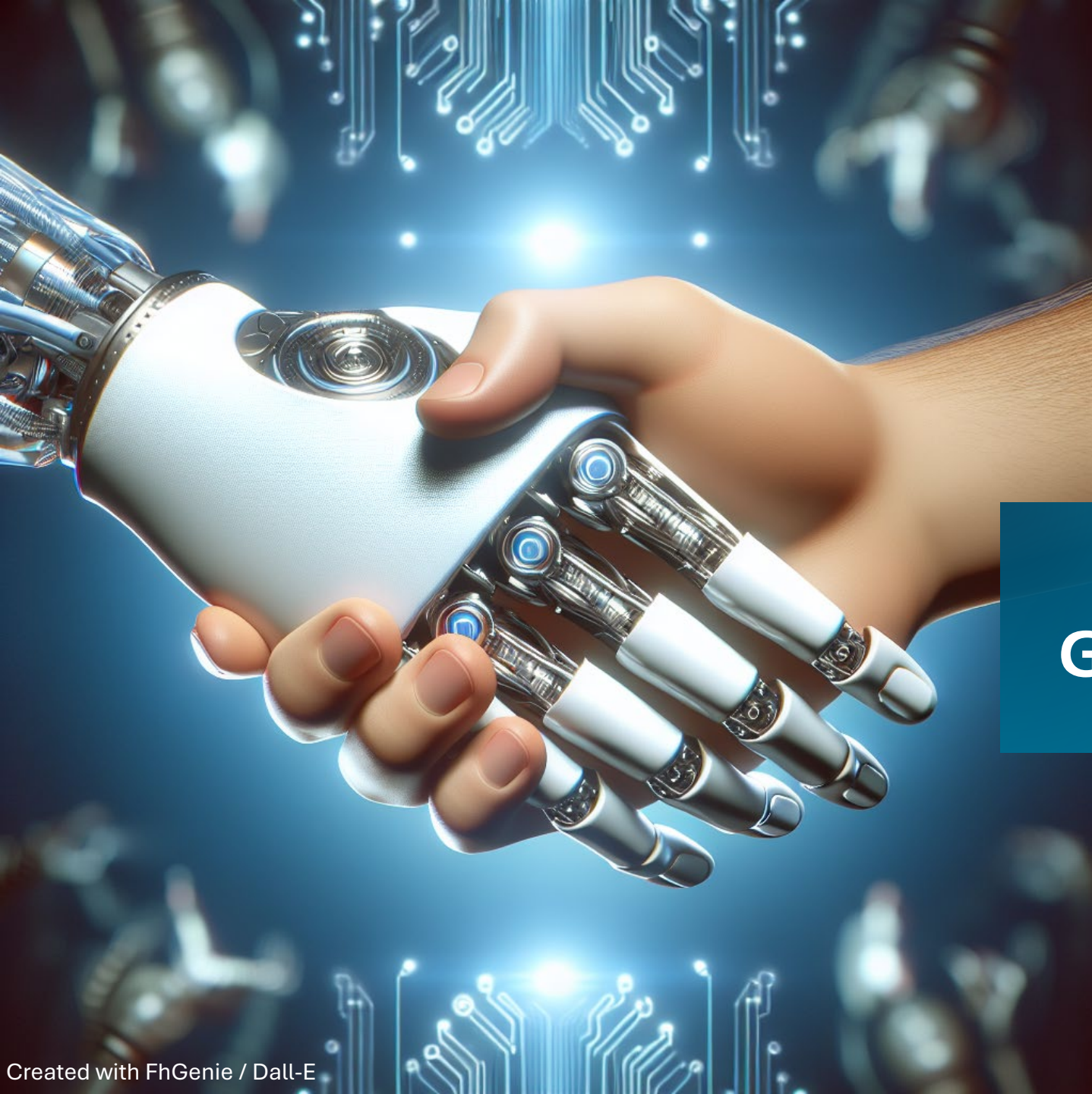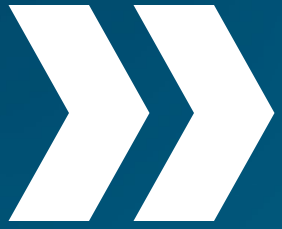
An LLM is a probabilistic model trained on extensive data to generate meaningful word sequences.

28.11.2024   © Fraunhofer   **Public**

Fraunhofer

There are several application fields for LLMs in SE

LLMs in SE

Requirements	Development	QA	…
▪ NLP for requirements ▪ Creation of personas ▪ Exploration of scenarios ▪ …	▪ Code generation ▪ Code review ▪ …	▪ Test case generation ▪ …	▪ …

Documentation generation

Chat bots / Assistants…

These are just a few examples!

28.11.2024   © Fraunhofer   **Public**

Imagine that System A must no longer concerned about B's schema

28.11.2024 © Fraunhofer **Public**
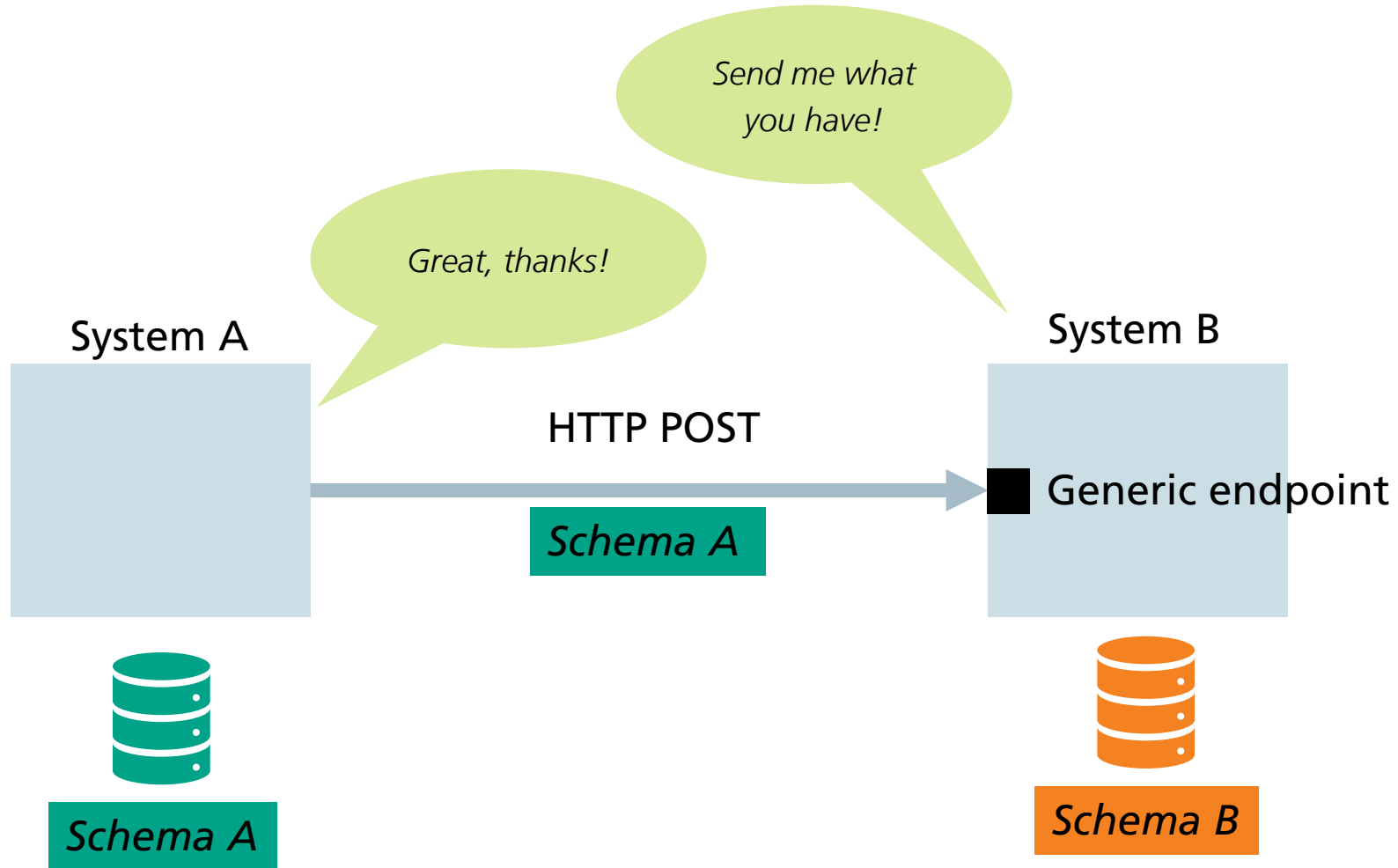
With the rise of LLMs, there is an open door for the design of "creative systems"— systems that can perform human-like tasks to achieve qualities such as interoperability at runtime.

Public
Fraunhofer

How does it look like for a customer record?

Schema X

```json
{
  "customerData": {
    "customerId": 67890,
    "name": {
      "first": "Jane",
      "last": "Smith"
    },
    "contactInfo": {
      "emailAddress": "janesmith@domain.com",
      "phone": "+1-555-9876"
    },
    "address": "200 Elm St, Apt 12A, Metropolis, NY, 10001, USA",
    "birthDate": "1990-09-23",
    "loyaltyTier": "Platinum",
    "signupDate": "2021-05-22"
  }
}
```
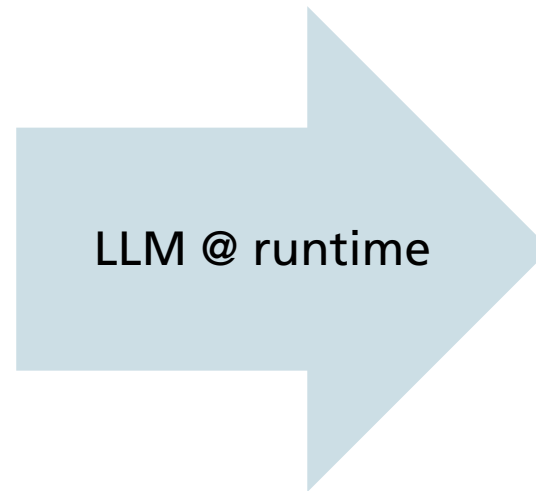
Schema Y

```json
{
  "customer": {
    "id": 12345,
    "firstName": "John",
    "lastName": "Doe",
    "email": "johndoe@example.com",
    "phoneNumber": "+1-555-1234",
    "address": {
      "streetName": "Main Street",
      "streetNumber": "100",
      "apartmentNumber": "5B",
      "city": "Springfield",
      "state": "IL",
      "zipCode": "62704",
      "country": "USA"
    },
    "dateOfBirth": "1985-06-15",
    "membershipLevel": "Gold",
    "registrationDate": "2022-01-10"
  }
}
```

How does it look like for a customer record?
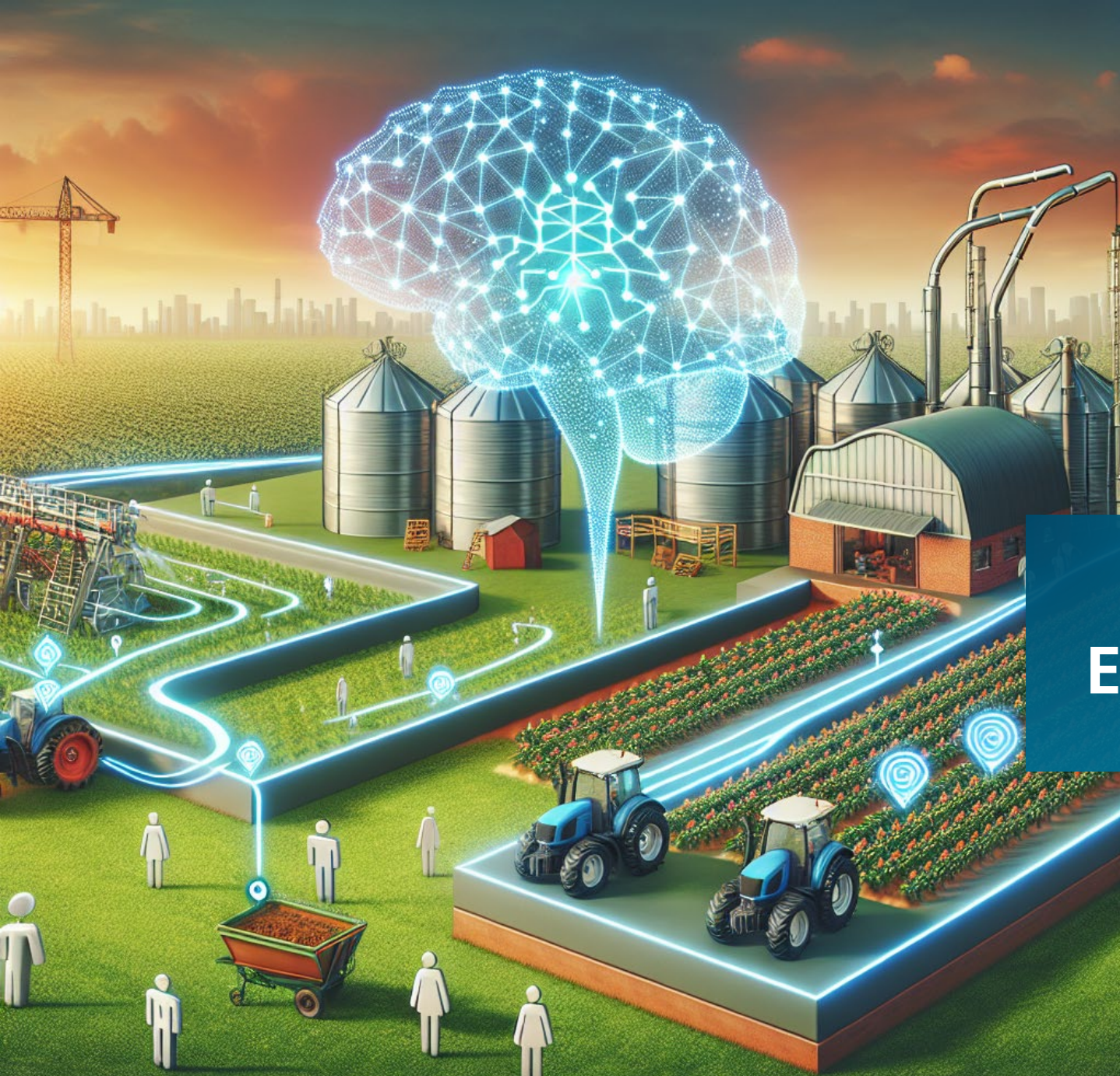
Schema X

```json
{
  "customerData": {
    "customerId": 67890,
    "name": {
      "first": "Jane",
      "last": "Smith"
    },
    "contactInfo": {
      "emailAddress": "janesmith@domain.com",
      "phone": "+1-555-9876"
    },
    "address": "200 Elm St, Apt 12A, Metropolis, NY, 10001, USA",
    "birthDate": "1990-09-23",
    "loyaltyTier": "Platinum",
    "signupDate": "2021-05-22"
  }
}
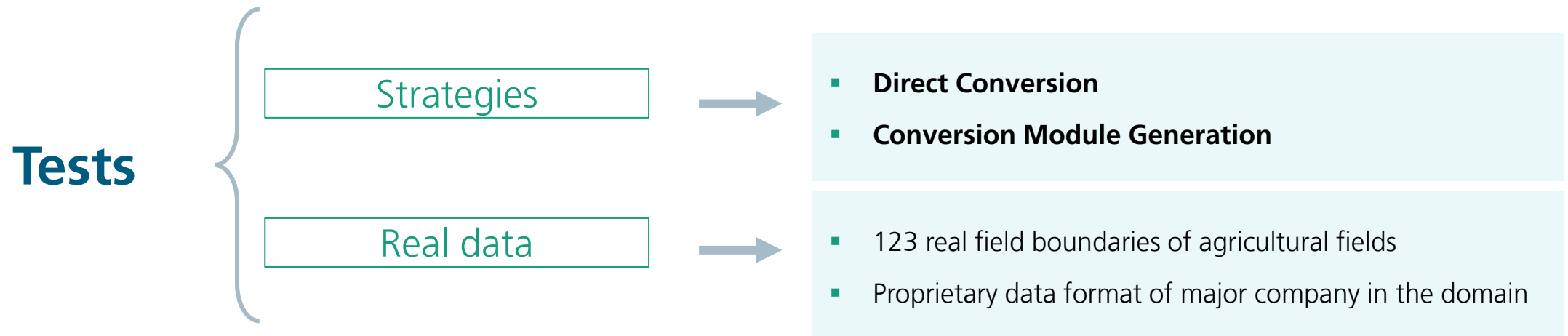```

LLM @ runtime

Schema Y

```json
{
  "customer": {
    "id": 67890,
    "firstName": "Jane",
    "lastName": "Smith",
    "email": "janesmith@domain.com",
    "phoneNumber": "+1-555-9876",
    "address": {
      "streetName": "Elm St",
      "streetNumber": "200",
      "apartmentNumber": "12A",
      "city": "Metropolis",
      "state": "NY",
      "zipCode": "10001",
      "country": "USA"
    },
    "dateOfBirth": "1990-09-23",
    "membershipLevel": "Platinum",
    "registrationDate": "2021-05-22"
```

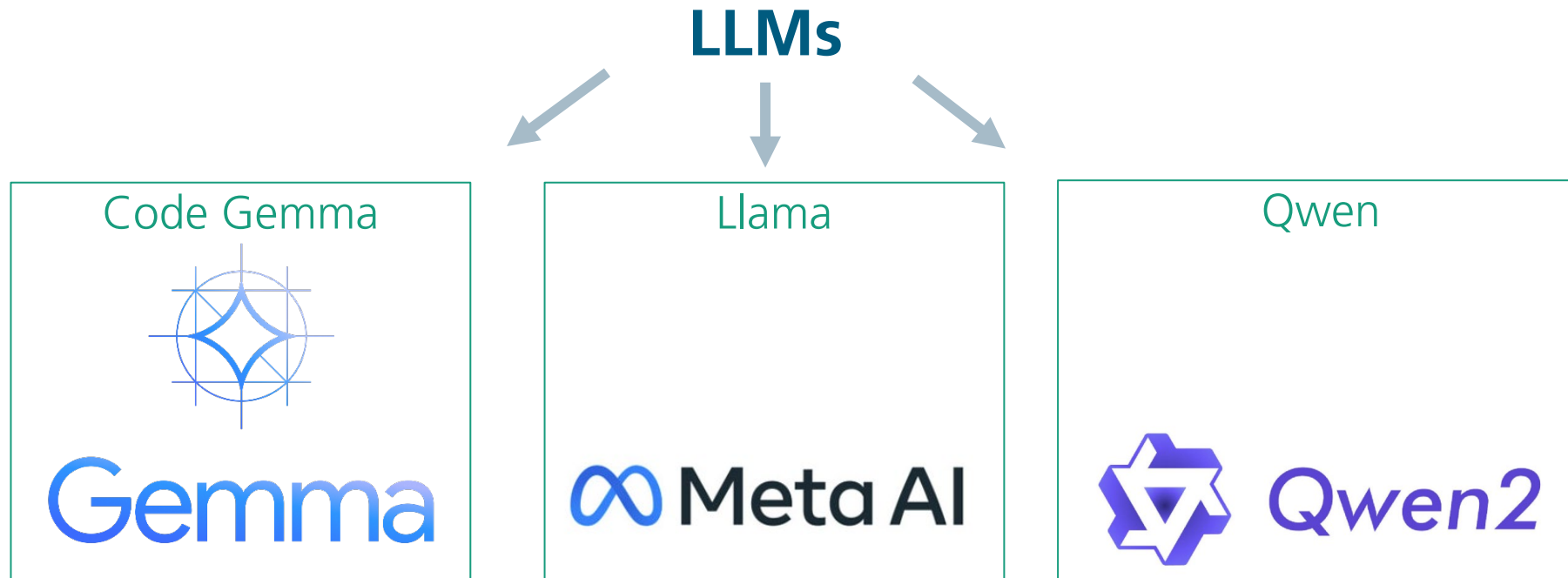Prompt: "Convert data from format: <input> to format: <output example>"

Fraunhofer

Experiences

Created with FhGenie / Dall-E

We tested two strategies to convert real field data from a proprietary schema

Tests

Strategies

- **Direct Conversion**
- **Conversion Module Generation**

Real data

- 123 real field boundaries of agricultural fields
- Proprietary data format of major company in the domain

Fraunhofer

We have tried out three LLMs to implement the adapters at runtime

LLMs

Code Gemma	Llama	Qwen

28.11.2024   © Fraunhofer   **Public**

We created a demonstrator to illustrate the strategy "direct conversion"
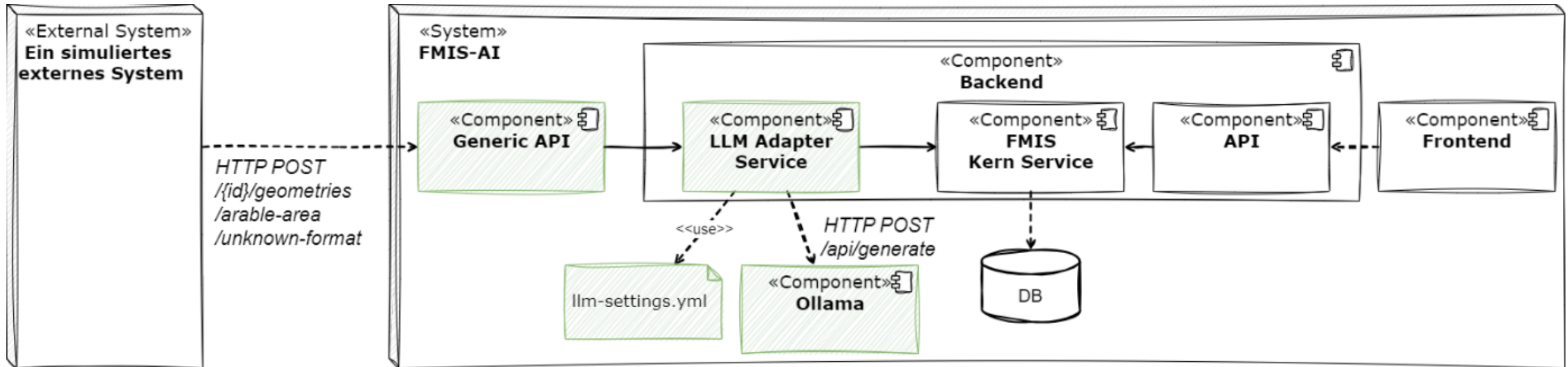
We used zero-shot prompting in our application

Prompting

| Zero-shot | → | Just prompt the models once to get the response |

Our approach

| Few-shot | → | Provide the models with a few examples before you go |

Not applicable

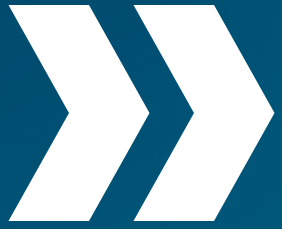| Fine-tuning* | → | Use datasets to refine base models |

Effort intensive

28.11.2024 © Fraunhofer **Public**

The results???

IT WORKS!
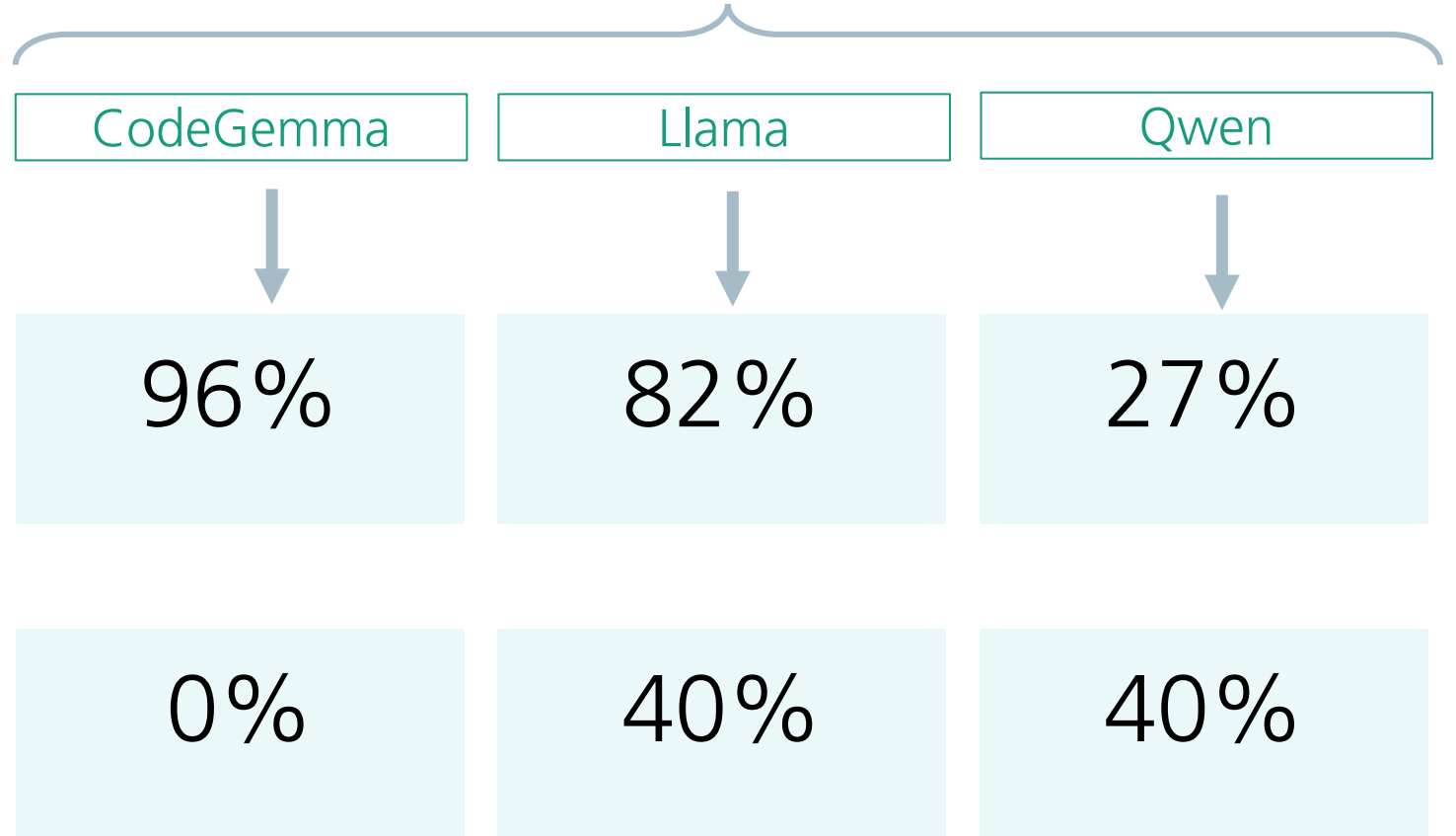
However, the details make the difference, as many design decisions must be made when setting up an LLM-based interface.

© Fraunhofer
Public
Fraunhofer

Some aspects are generic, while others are use-case specific
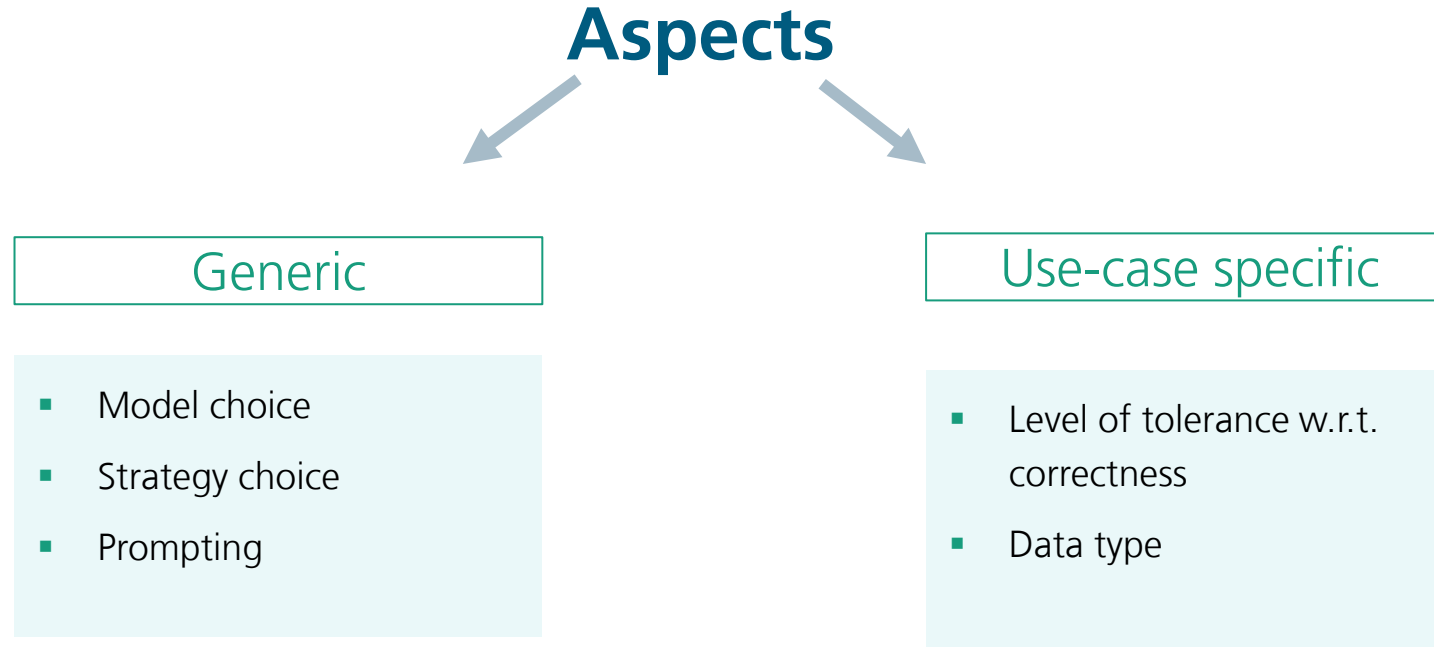
Aspects

Generic

- Model choice
- Strategy choice
- Prompting

Use-case specific

- Level of tolerance w.r.t. correctness
- Data type

Fraunhofer

We tested models from the families Gemma, Llama, and Qwen using two strategies

Models

	CodeGemma	Llama	Qwen
Direct Conversion	96%	82%	27%
Conversion Module	0%	40%	40%

Strategy

Fraunhofer

Who's the best?

We tested models from the families Gemma, Llama, and Qwen using two strategies

Models

	CodeGemma	Llama	Qwen
Direct Conversion	96%	82%	27%
Conversion Module	0%	40%	40%

Strategy

But non-deterministic

Deterministic, though

Fraunhofer
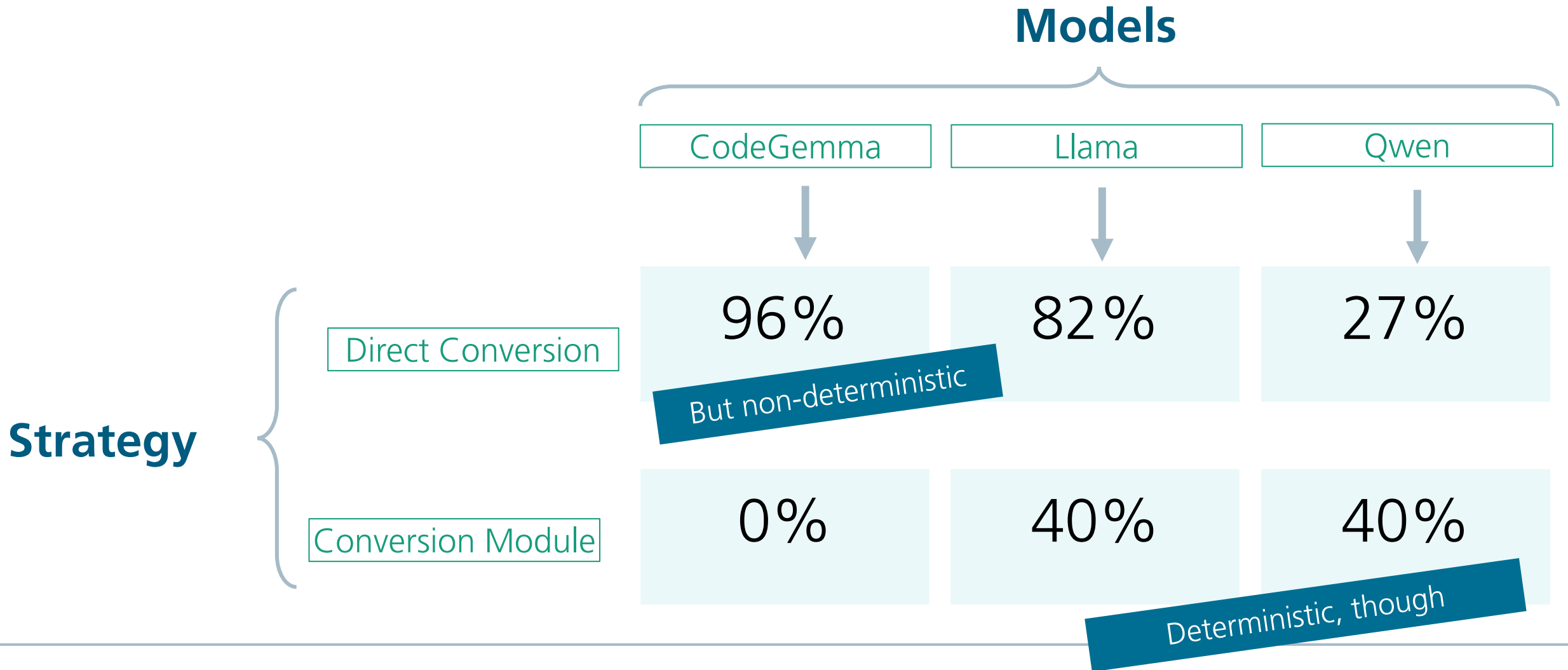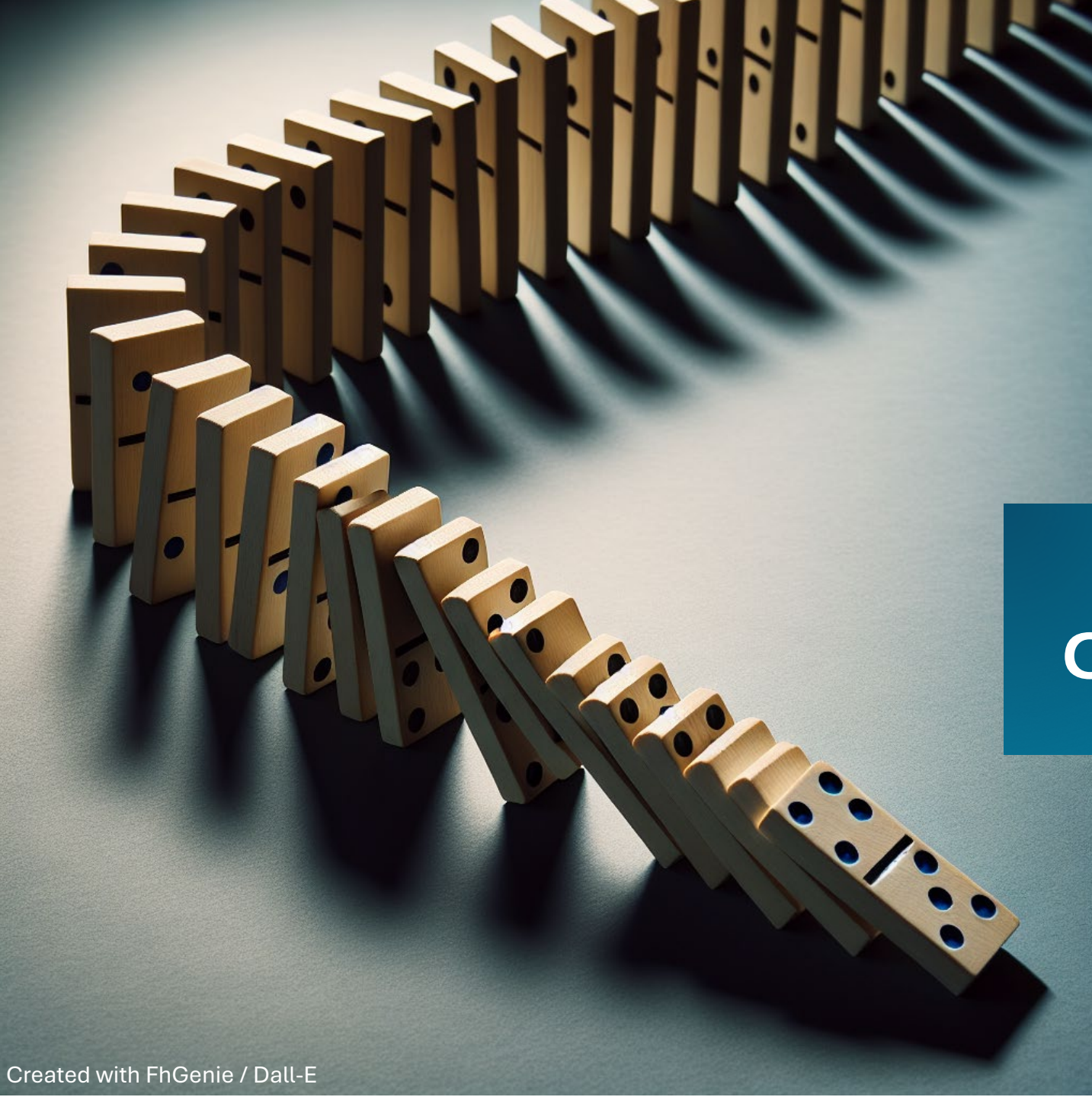
Consequences

It is possible to implement effective APIs for data conversion with LLM-based strategies

[Generic] APIs

"Effective" →
- 96%; can it be considered effective enought in my use case?
- Deterministic solutions can be generated

LLM-based strategies →
- Direct Conversion
- Conversion Module Generation
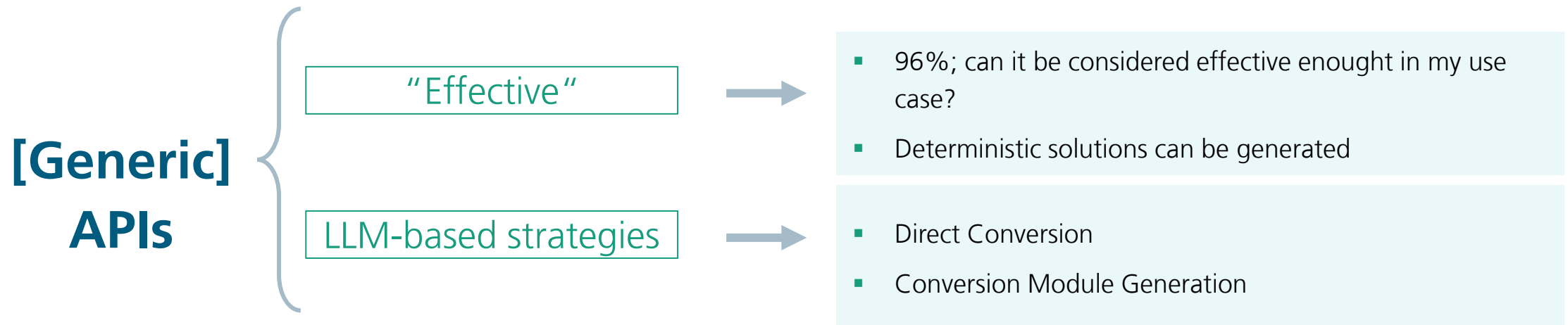
This can save integration efforts, reduce time to market, facilitate the onboarding of partners in your digital ecosystem, enable dynamic integration in open environments, etc.

Fraunhofer

No "one-size fits all"

Software engineers are required!

What's next?

Created with FhGenie / Dall-E

Next steps includes implementation, expansion, and application in industry

Next steps

Implementation → Our current demonstrator implements the strategy "Direct Conversion", we plan the implementation of "Conversion Module Generation" as well

Expansion → We have focused on the agricultural domain because we have developed several projects in the domain in recent years; Other domains are being planned (e.g., smart cities)

Industry → We are already in contact with industry partners to help them pioneer on the engineering of LLM-based APIs

Fraunhofer

Thank you for your attention!

Stay in touch!

Dr. Rodrigo Falcão

Fraunhofer IESE

✉ rodrigo.falcao@iese.fraunhofer.de

linkedin.com/in/rmfalcao