ACCELERATING THE ADOPTION OF ASSET ADMINISTRATION SHELLS THROUGH AI AGENTS

DR. CAMILO VELÁZQUEZ RODRÍGUEZ (camilo.velazquez@flandersmake.be)

STEVEN KAUFFMANN

TOM PAUWAERT

STIJN HUYSENTRUYT

AXL VAN ALBOOM

The Fifteenth International Conference on Intelligent Systems and Applications (INTELLI) 2026

March 08 - 12, 2026

Valencia, Spain

ABOUT ME

- Research Engineer, Flanders Make (Kortrijk, Belgium) — Digital Productions, Productions Core Lab.

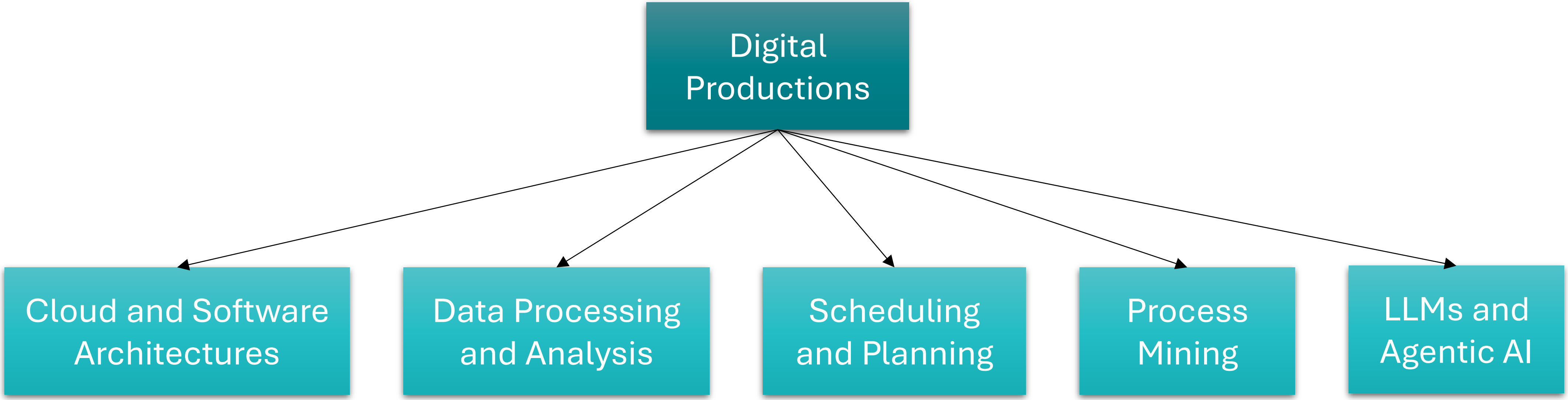
- Leads development of data-driven industrial solutions (e.g., data engineering, analytics, software and cloud architecture).

- Ph.D. (2024), Vrije Universiteit Brussel (VUB); Postdoctoral Researcher at the Software Languages Lab.

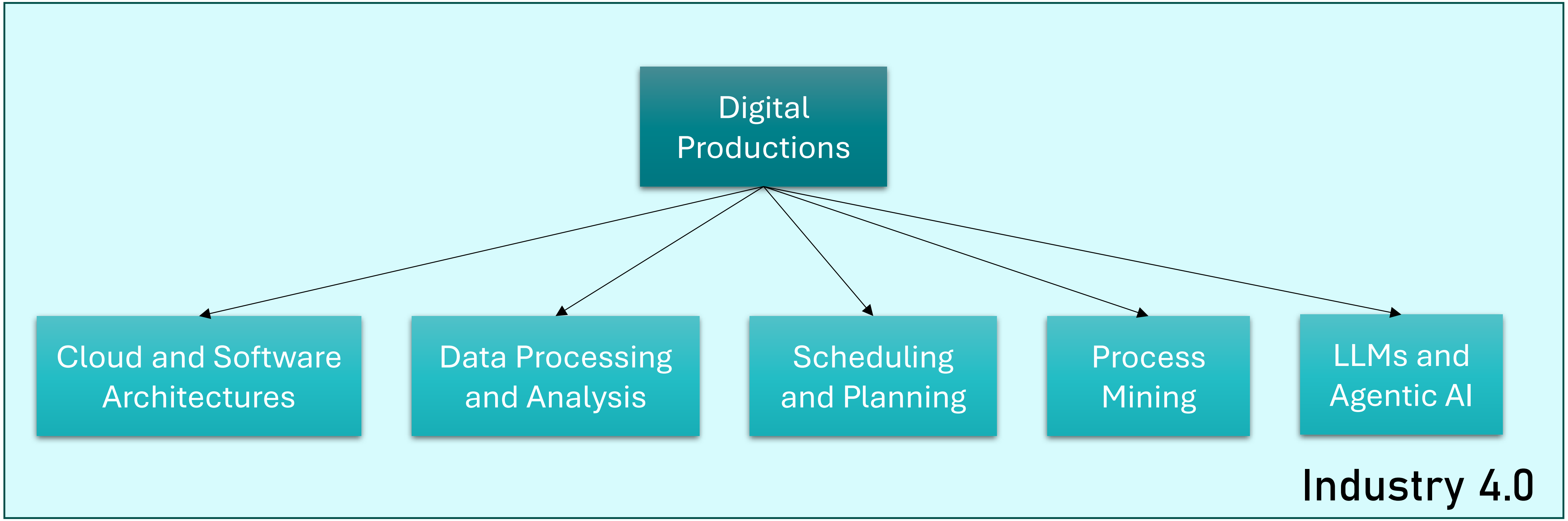
- Expertise in optimisation, machine and deep learning, and mathematical modelling.

- Focused on scalable, robust solutions for complex industrial systems.

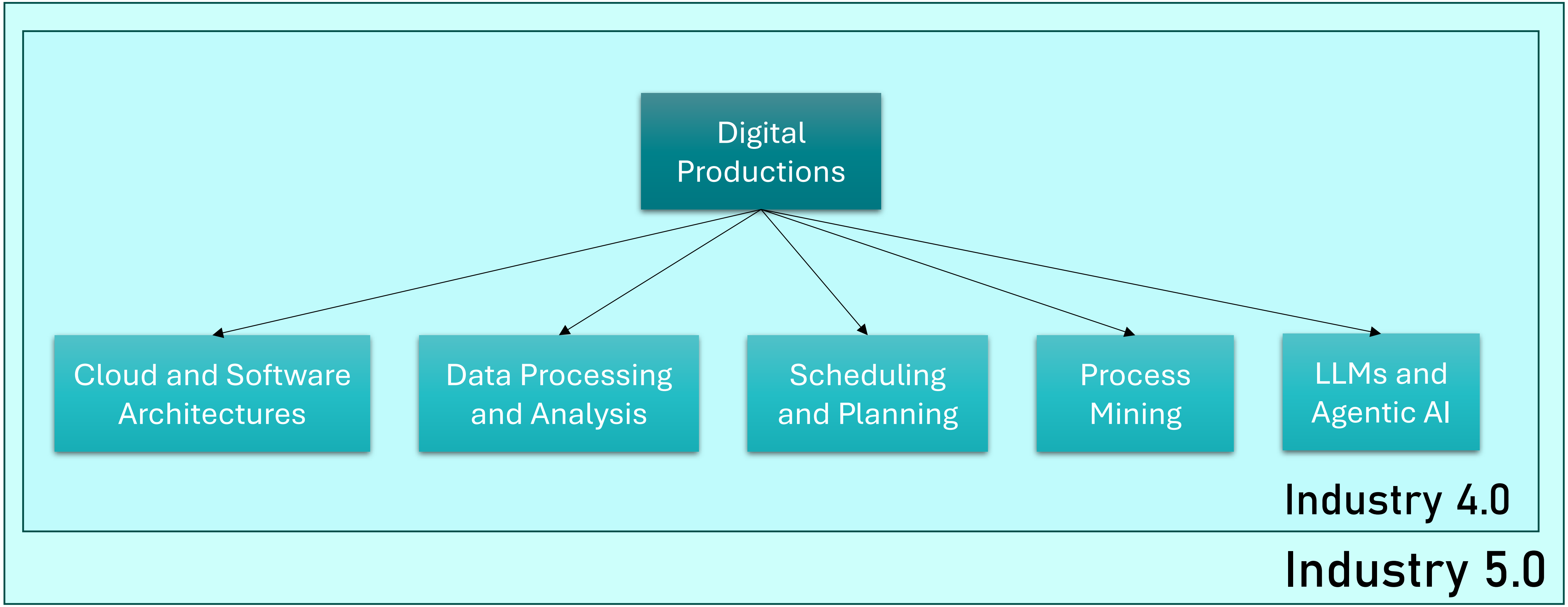
RESEARCH INTERESTS OF MY GROUP



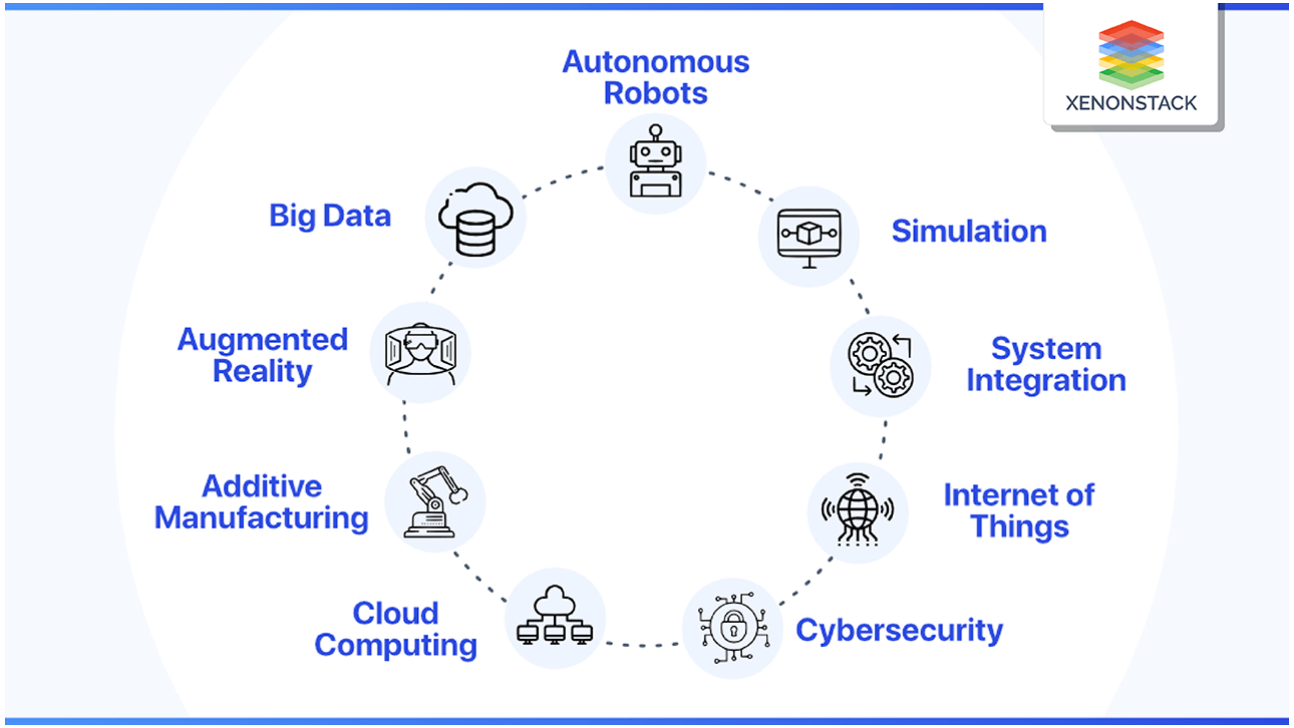
RESEARCH INTERESTS OF MY GROUP

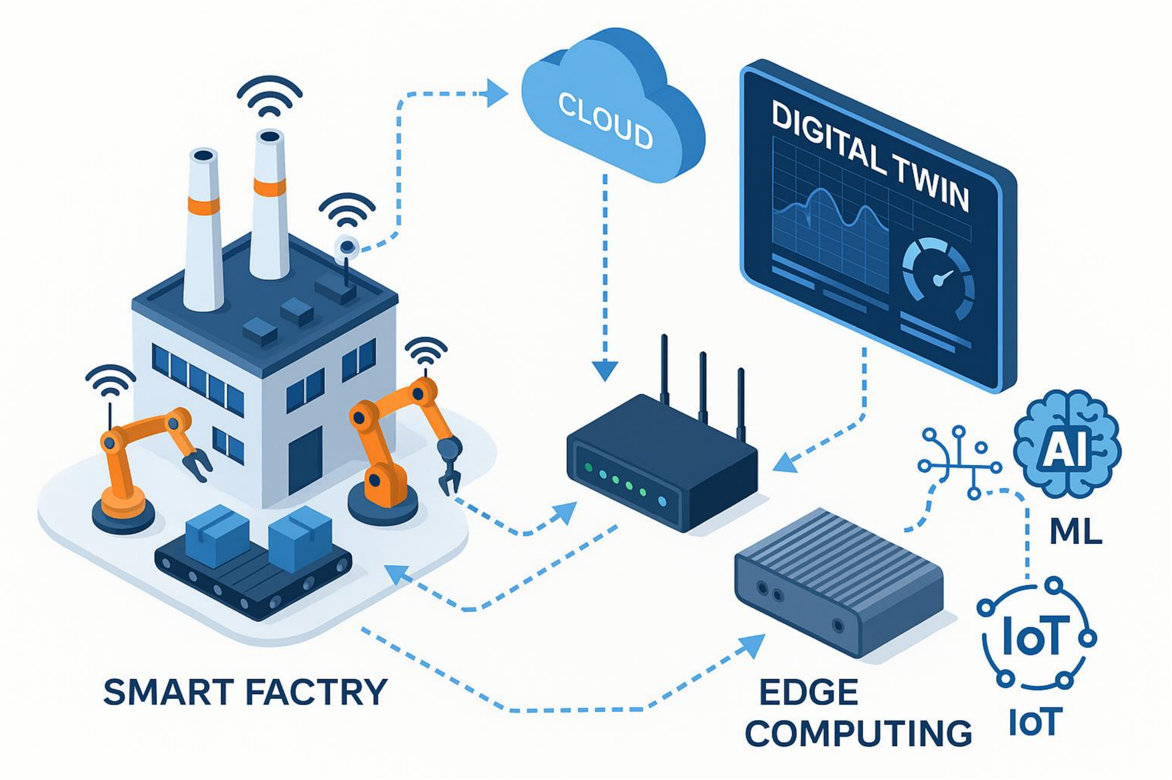


RESEARCH INTERESTS OF MY GROUP



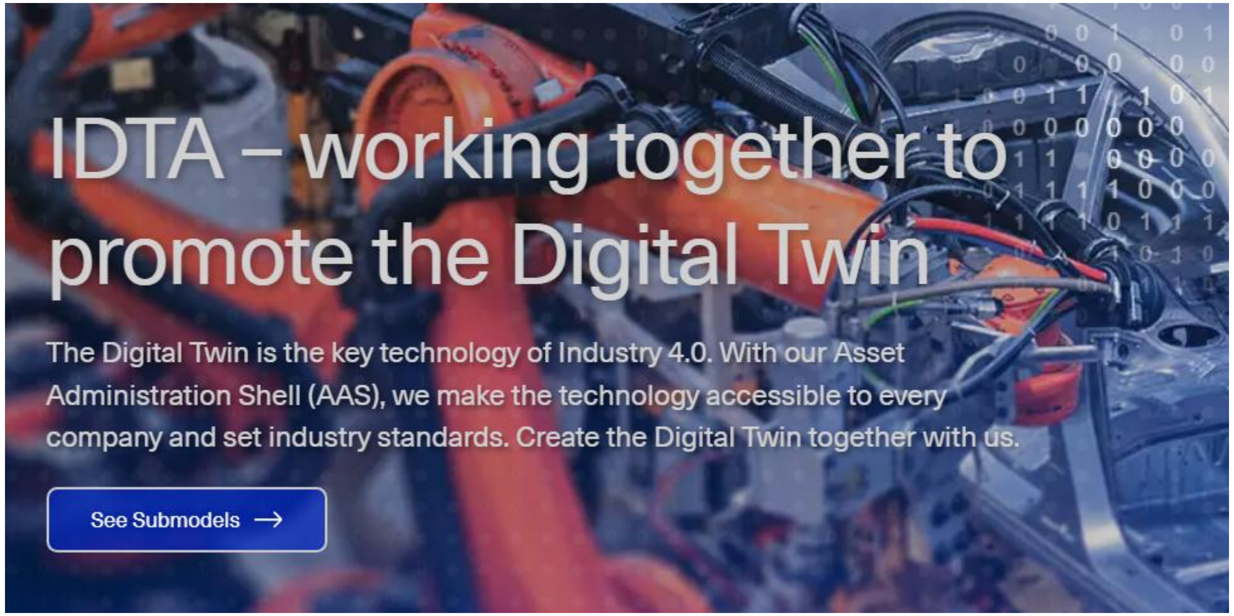
INDUSTRY 4.0 PARADIGM



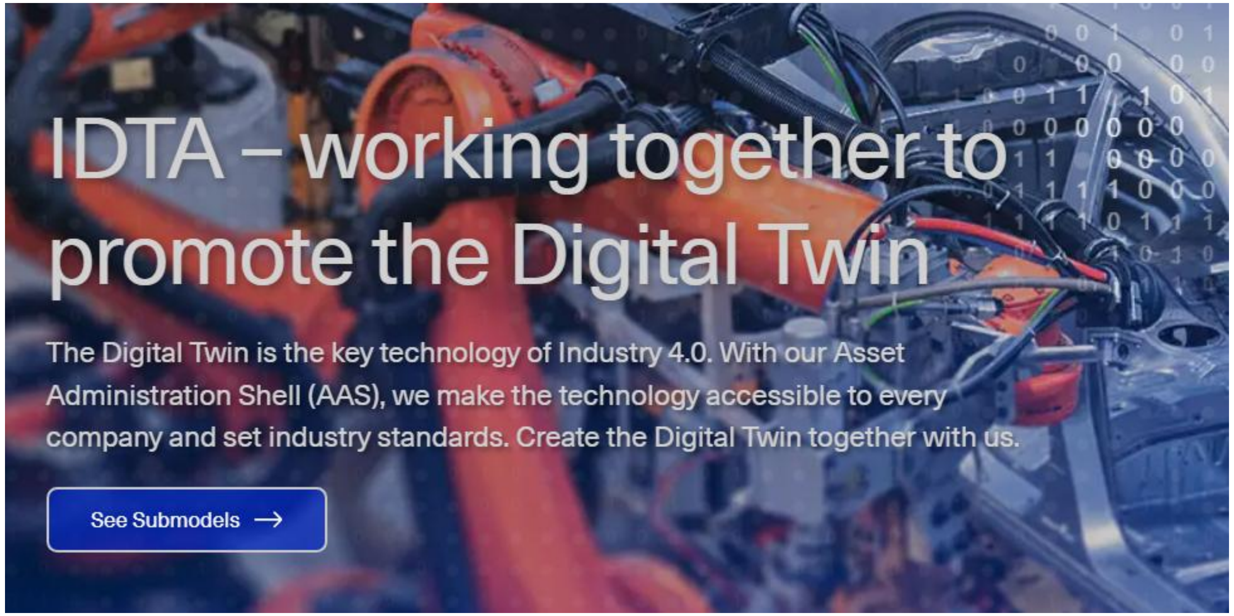


https://www.xenonstack.com/insights/digital-twin-in-manufacturing
https://inhandgo.com/en-ad/blogs/articles/revolutionizing-industry-4-0-how-digital-twins-are-powering-the-future-of-smart-manufacturing

ASSET ADMINISTRATION SHELLS (AAS)



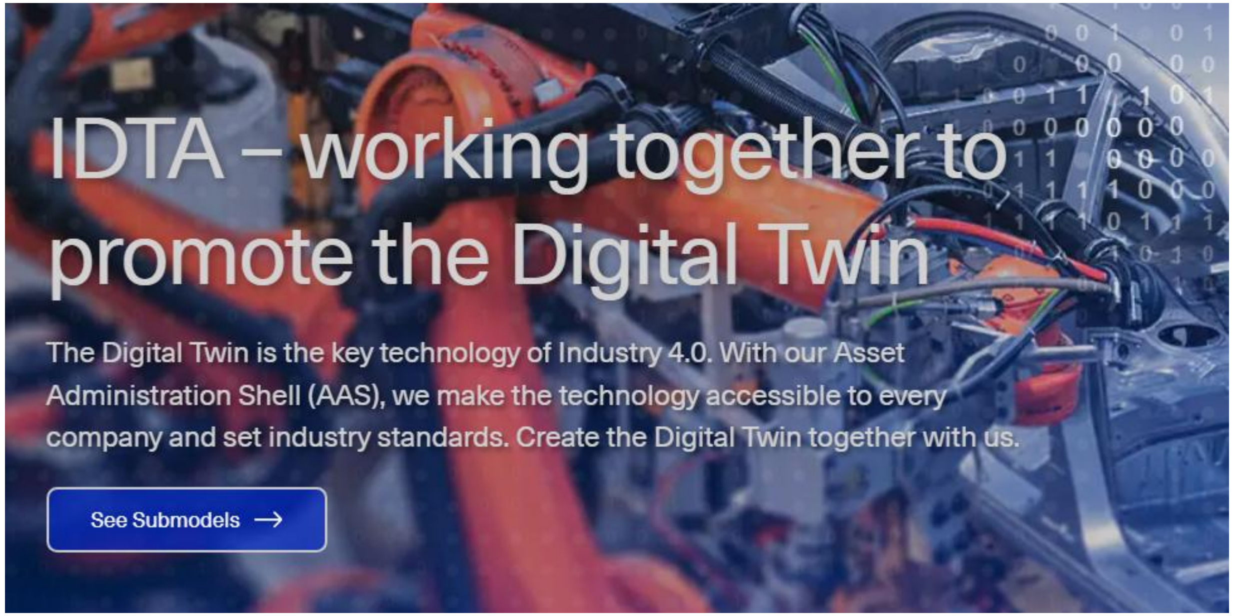
ASSET ADMINISTRATION SHELLS (AAS)



- Digital Nameplate for Industrial Equipment
- Functional Safety
- Technical Data for Industrial Equipment in Manufacturing

https://industrialdigitaltwin.org/en/

ASSET ADMINISTRATION SHELLS (AAS)



AAS Submodel Templates

Submodels constitute the content of the Asset Administration Shell. They describe content-related or functional aspects of an asset. Find the overview of the official IDTA submodel templates here.

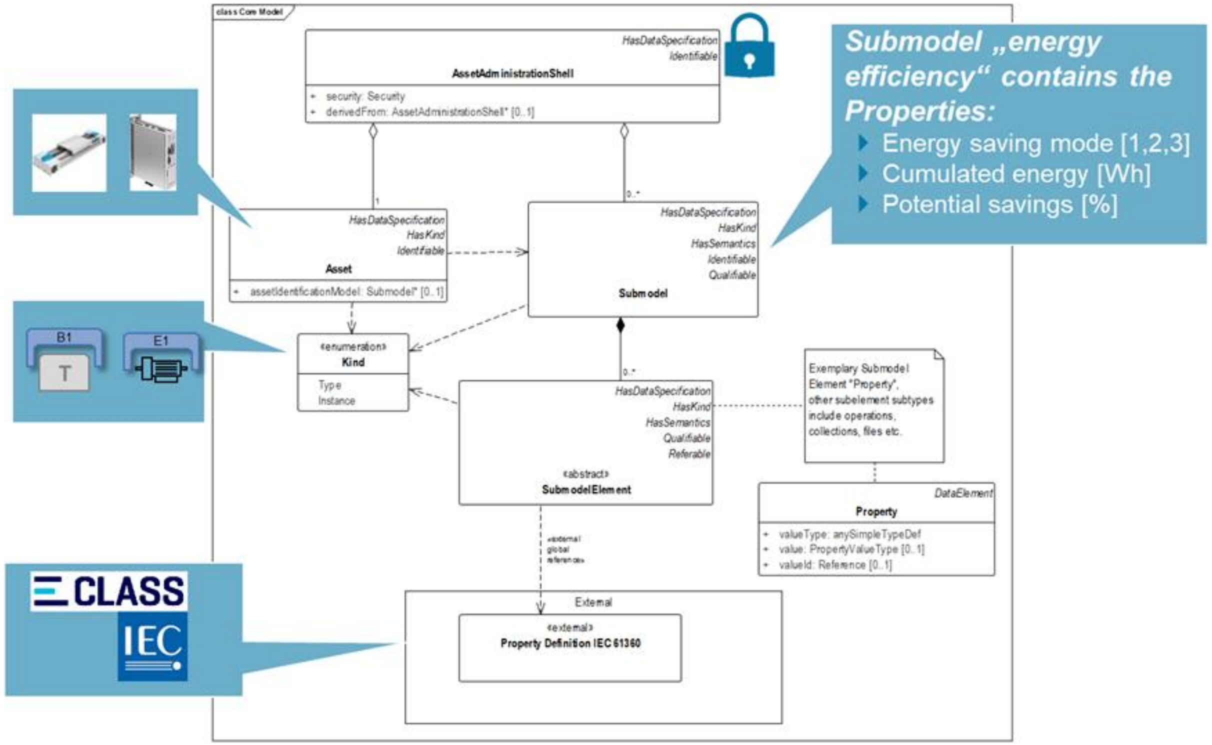
Registered AAS Submodel Templates

Number of our submodels: 100

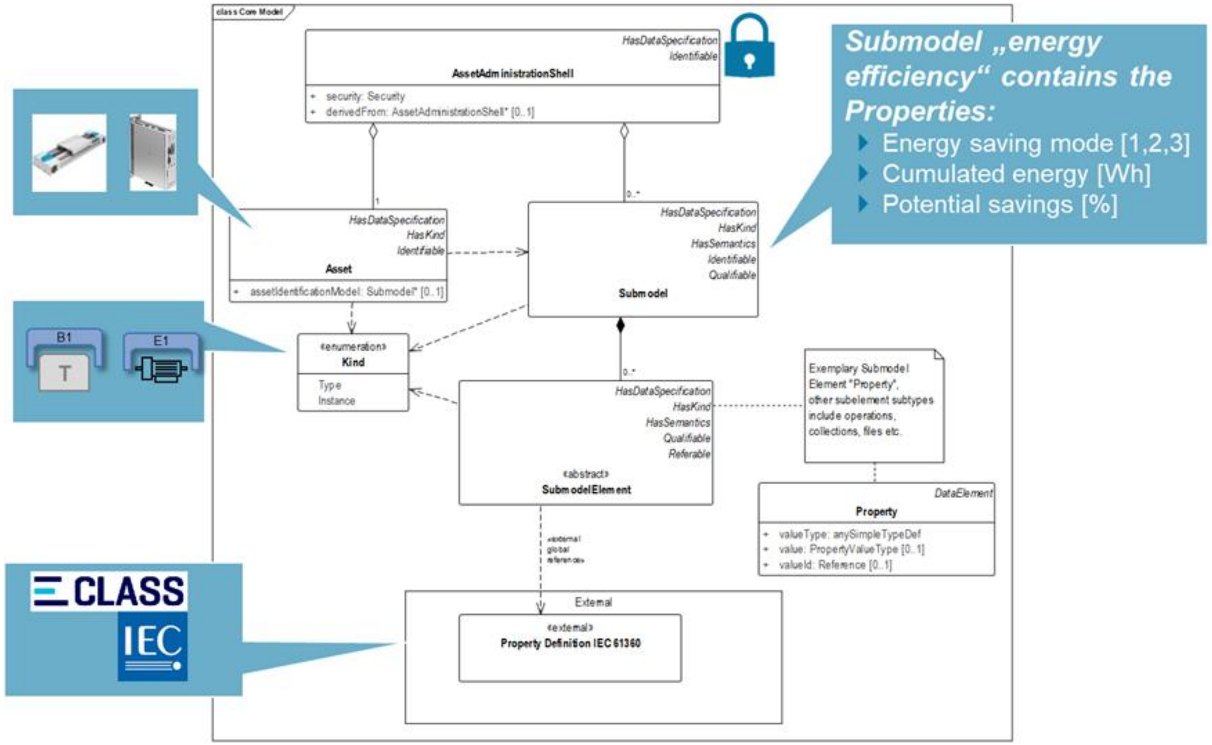
- Digital Nameplate for Industrial Equipment
- Functional Safety
- Technical Data for Industrial Equipment in Manufacturing

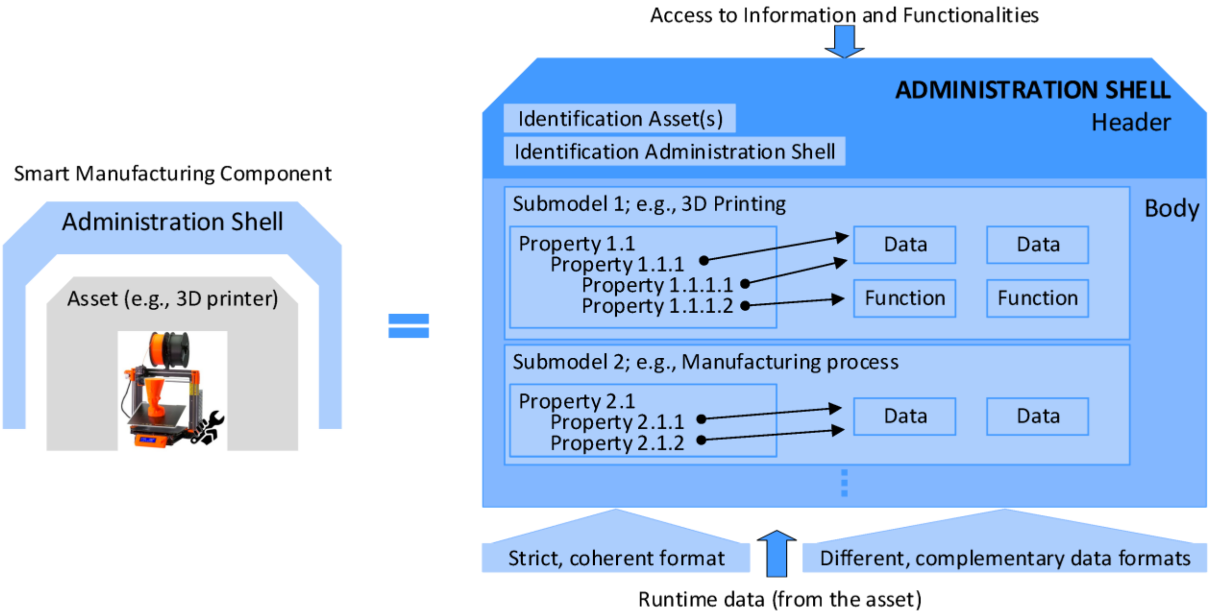
https://industrialdigitaltwin.org/en/

ASSET ADMINISTRATION SHELLS (AAS)



ASSET ADMINISTRATION SHELLS (AAS)





WHY ARE AASs RELEVANT?

WHY ARE AASs RELEVANT?

Predictive Maintenance

WHY ARE AASs RELEVANT?

Predictive Maintenance

Plug-and-produce

WHY ARE DIGITAL ASSETS RELEVANT?

Predictive Maintenance

Plug-and-produce

Traceability and Sustainability

WHAT ARE THE CURRENT LIMITATIONS?

Manual Labour

WHAT ARE THE CURRENT LIMITATIONS?

Manual Labour

Lack of Expertise

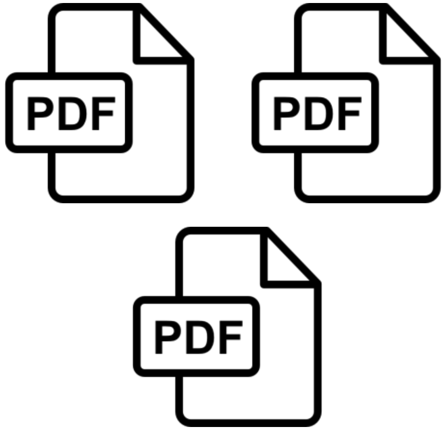
WHAT ARE THE CURRENT LIMITATIONS?

Manual Labour

Lack of Expertise

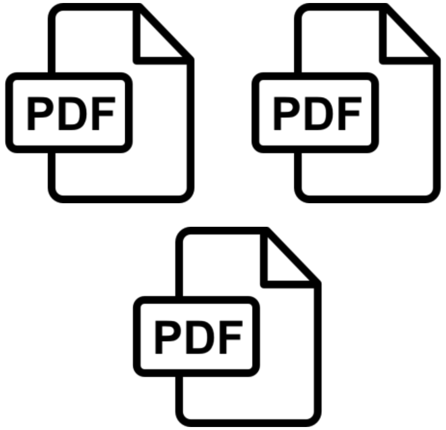
Lack of Awarness

AUTOMATIC INFORMATION EXTRACTION

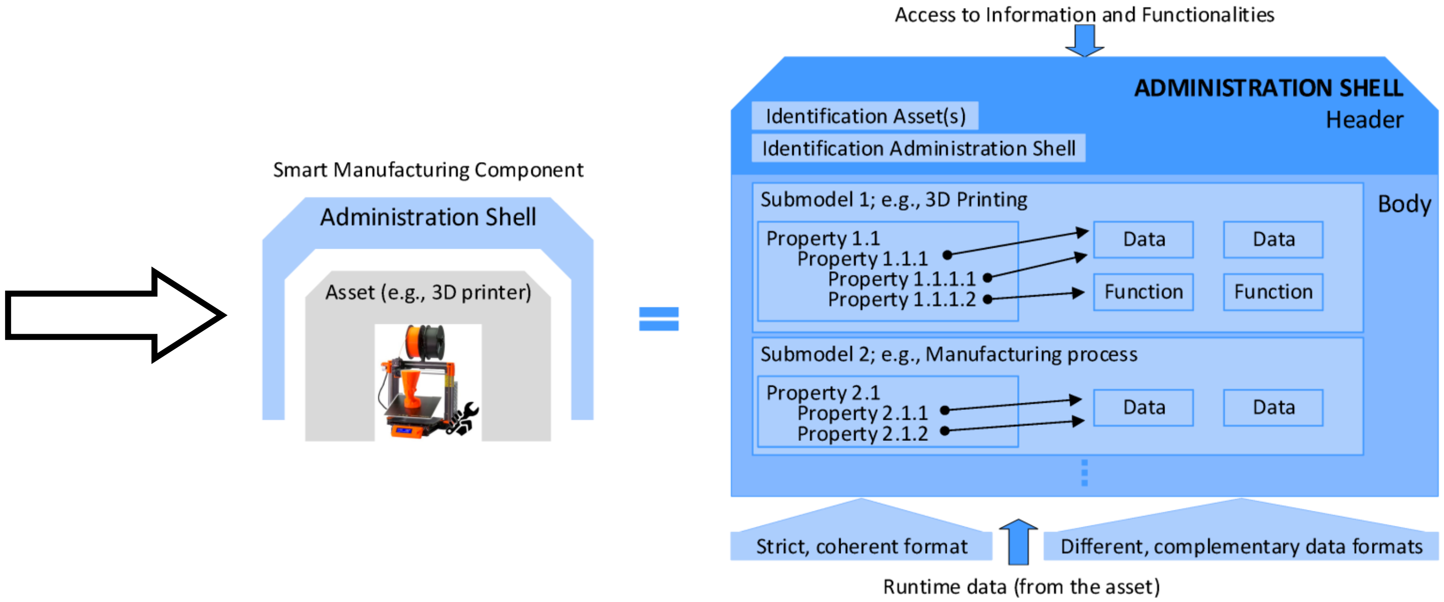


PDF Documents

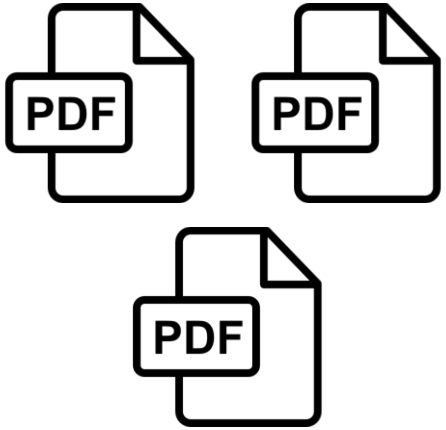
AUTOMATIC INFORMATION EXTRACTION



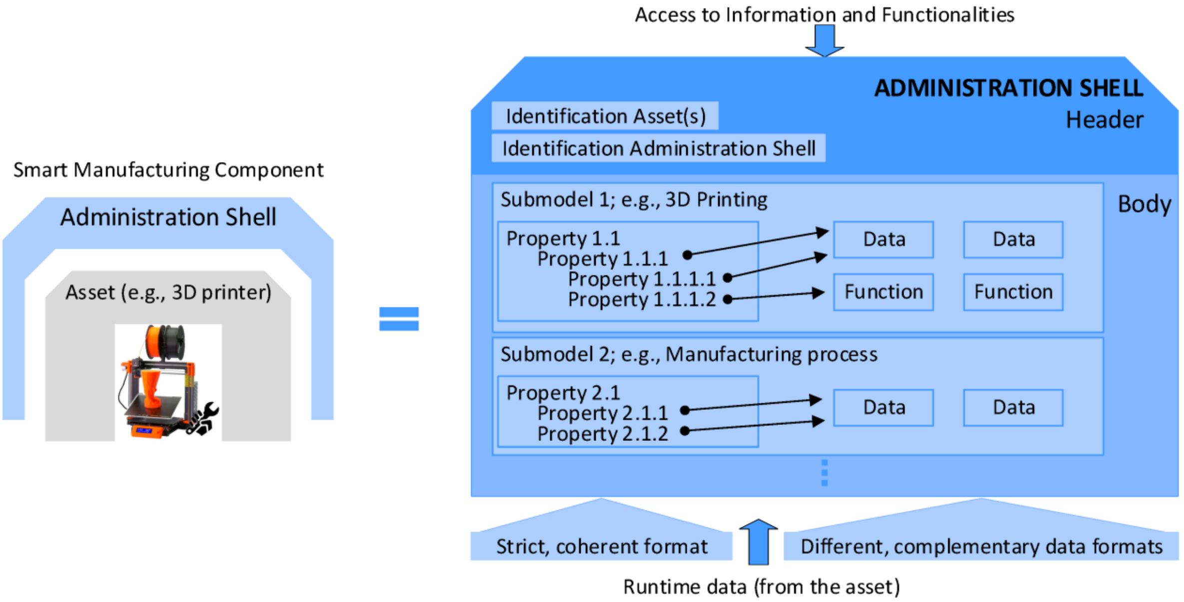
PDF Documents



AUTOMATIC INFORMATION EXTRACTION

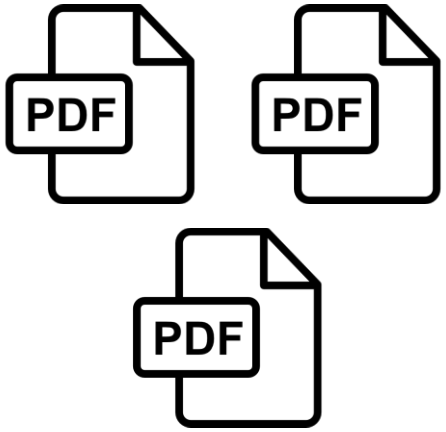


PDF Documents

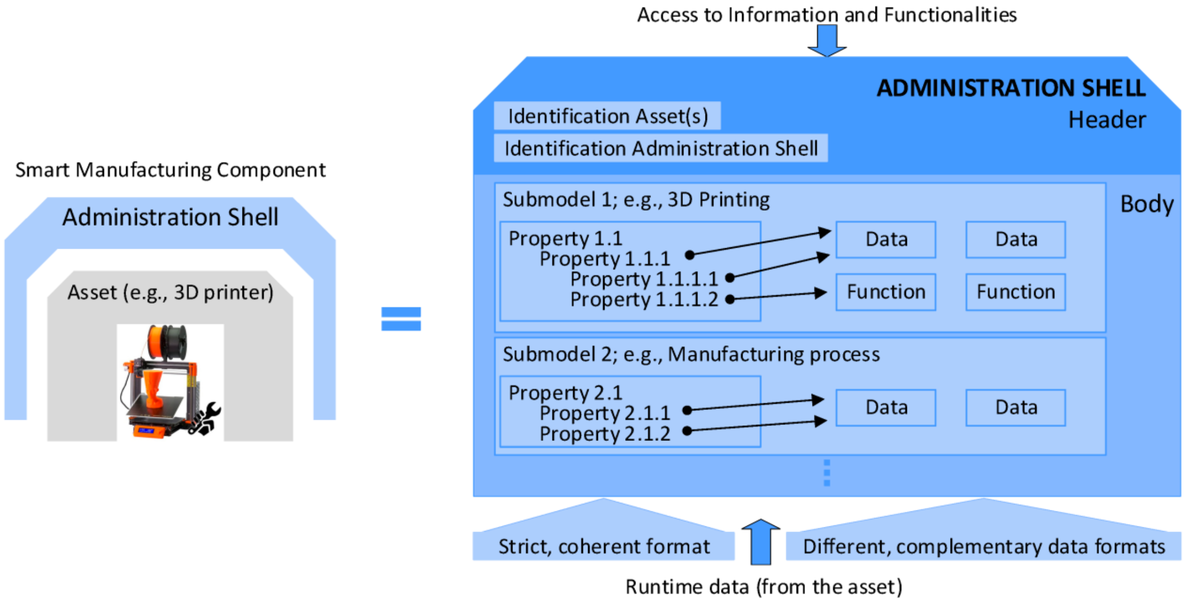


Heterogeneous
Sources

AUTOMATIC INFORMATION EXTRACTION

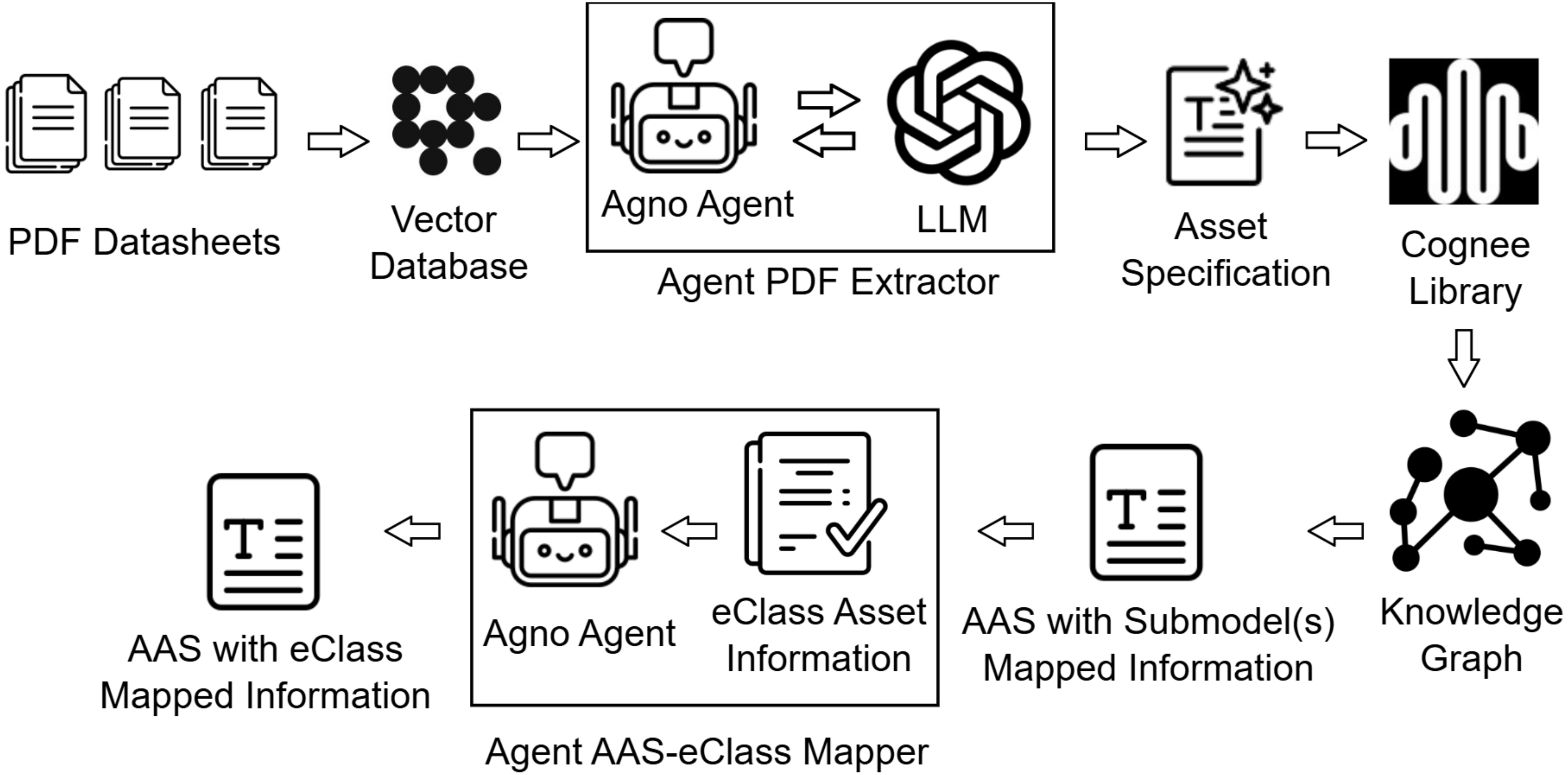


PDF Documents



Heterogeneous Sources ⇒ Homogeneous/Standard Outcome

OUR SOLUTION



SEMANTIC EQUIVALENCE

Characteristics

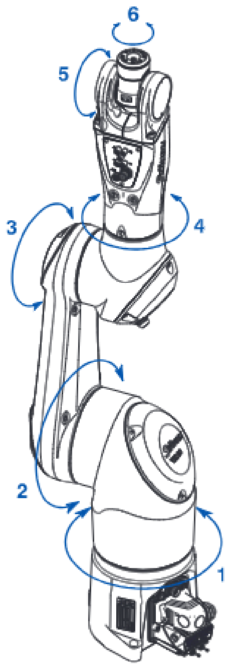
	TX2-90	TX2-90L	TX2-90XL
Load capacity	14 kg	12 kg	7 kg
Reach (between axis 1 and 6)	1000 mm	1200 mm	1450 mm
Number of degrees of freedom	6	6	6
Repeatability - ISO 9283	± 0.02 mm	± 0.02 mm	± 0.02 mm
Weight	114 kg	117 kg	119 kg
UL certification	✓	✓	✓
Attachment methods	360°		

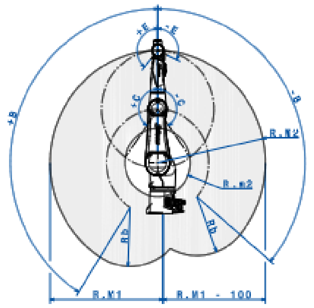
Performance

Joint speed -axis 1	330°/s	345°/s	330°/s
Joint speed -axis 2	340°/s	340°/s	350°/s
Joint speed -axis 3	430°/s	420°/s	410°/s
Joint speed -axis 4	540°/s	540°/s	540°/s
Joint speed -axis 5	475°/s	475°/s	475°/s
Joint speed -axis 6	760°/s	760°/s	760°/s
Maximum speed at load gravity center	10.9 m/s	11.1 m/s	11.6 m/s
Maximum inertia axis 5	1.5 kg.m²	1.25 kg.m²	1 kg.m²
Maximum inertia axis 6	0.25 kg.m²	0.20 kg.m²	0.15 kg.m²
Brakes	All axes		

Work envelope and range of motion

Maximum reach between axis 1 and 5 (R. M)	900 mm	1100 mm	1350 mm
Minimum reach between axis 1 and 5 (R.m1)	200 mm	272 mm	327 mm
Minimum reach between axis 2 and 5 (R.m2)	256 mm	320 mm	391 mm
Reach between axis 3 and 5 (R.b)	425 mm	550 mm	650 mm





SEMANTIC EQUIVALENCE

Characteristics

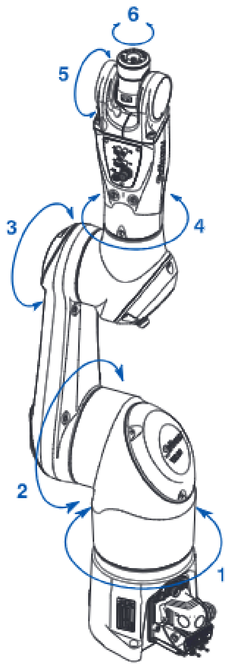
	TX2-90	TX2-90L	TX2-90XL
Load capacity	14 kg	12 kg	7 kg
Reach (between axis 1 and 6)	1000 mm	1200 mm	1450 mm
Number of degrees of freedom	6	6	6
Repeatability - ISO 9283	± 0.02 mm	± 0.02 mm	± 0.02 mm
Weight	114 kg	117 kg	119 kg
UL certification	✓	✓	✓
Attachment methods	360°		

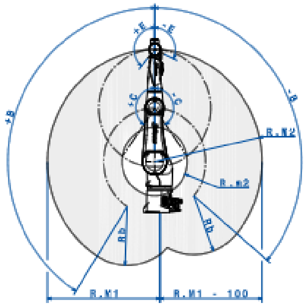
Performance

	TX2-90	TX2-90L	TX2-90XL
Joint speed -axis 1	330°/s	345°/s	330°/s
Joint speed -axis 2	340°/s	340°/s	350°/s
Joint speed -axis 3	430°/s	420°/s	410°/s
Joint speed -axis 4	540°/s	540°/s	540°/s
Joint speed -axis 5	475°/s	475°/s	475°/s
Joint speed -axis 6	760°/s	760°/s	760°/s
Maximum speed at load gravity center	10.9 m/s	11.1 m/s	11.6 m/s
Maximum inertia axis 5	1.5 kg.m²	1.25 kg.m²	1 kg.m²
Maximum inertia axis 6	0.25 kg.m²	0.20 kg.m²	0.15 kg.m²
Brakes	All axes		

Work envelope and range of motion

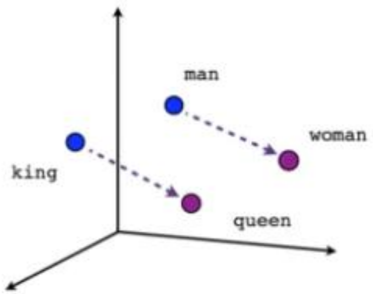
	TX2-90	TX2-90L	TX2-90XL
Maximum reach between axis 1 and 5 (R. M)	900 mm	1100 mm	1350 mm
Minimum reach between axis 1 and 5 (R.m1)	200 mm	272 mm	327 mm
Minimum reach between axis 2 and 5 (R.m2)	256 mm	320 mm	391 mm
Reach between axis 3 and 5 (R.b)	425 mm	550 mm	650 mm



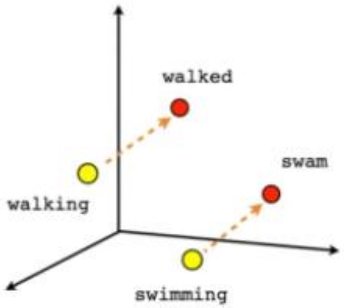


- AAS "DigitalNameplateAAS" [https://admin-shell.io/idta/aas/DigitalNameplate/3/0] of [https://admin-shell.io/idta/asset/DigitalName
  - Asset AssetInformation https://admin-shell.io/idta/asset/DigitalNameplate/3/0
  - SM <T> "Nameplate" V3.0 [https://admin-shell.io/idta/SubmodelTemplate/DigitalNameplate/3/0]
    - Prop "URIOfTheProduct" = https://www.domain-abc.com/Model-Nr-1234/Serial-Nr-5678 @{SMT/Cardinality=One}
    - MLP "ManufacturerName" → "Muster AG" @{SMT/Cardinality=One}
    - MLP "ManufacturerProductDesignation" → "ABC-123" @{SMT/Cardinality=One}
    - SMC "AddressInformation" @{SMT/Cardinality=One}
    - MLP "ManufacturerProductRoot" → "flow meter" @{SMT/Cardinality=ZeroToOne}
    - MLP "ManufacturerProductFamily" → "Type ABC" @{SMT/Cardinality=ZeroToOne}
    - Prop "ManufacturerProductType" = FM-ABC-1234 @{SMT/Cardinality=ZeroToOne}
    - Prop "OrderCodeOfManufacturer" = FMABC1234 @{SMT/Cardinality=One}
    - Prop "ProductArticleNumberOfManufacturer" = FM11-ABC22-123456 @{SMT/Cardinality=ZeroToOne}
    - Prop "SerialNumber" = 12345678 @{SMT/Cardinality=ZeroToOne}
    - Prop "YearOfConstruction" = 2022 @{SMT/Cardinality=ZeroToOne}
    - Prop "DateOfManufacture" = 2022-01-01 @{SMT/Cardinality=ZeroToOne}
    - Prop "HardwareVersion" = 1.0.0 @{SMT/Cardinality=ZeroToOne}
    - Prop "FirmwareVersion" = 1.0.0 @{SMT/Cardinality=ZeroToOne}
    - Prop "SoftwareVersion" = 1.0.0 @{SMT/Cardinality=ZeroToOne}
    - Prop "CountryOfOrigin" = DE @{SMT/Cardinality=ZeroToOne}
    - Prop "UniqueFacilityIdentifier" = 987654321 @{SMT/Cardinality=ZeroToOne}
    - File "CompanyLogo" @{SMT/Cardinality=ZeroToOne}
    - SML "Markings" (1 elements) @{SMT/Cardinality=ZeroToOne}
    - SMC "AssetSpecificProperties" (4 elements) @{SMT/Cardinality=ZeroToOne}

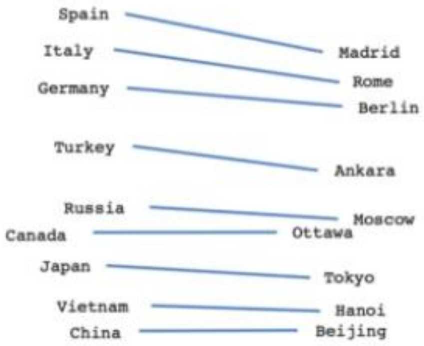
SEMANTIC EQUIVALENCE



Male-Female

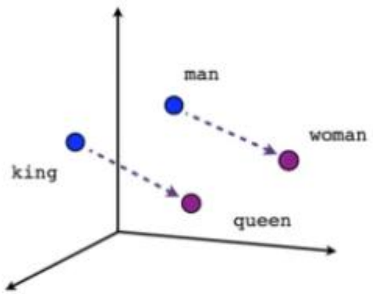


Verb tense

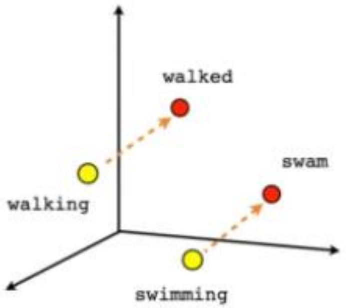


Country-Capital

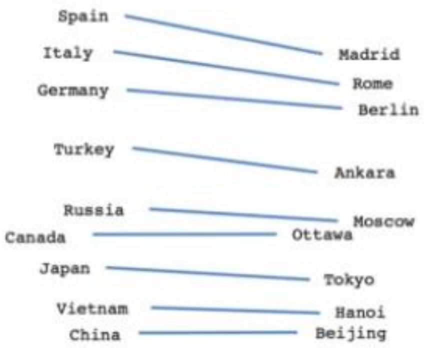
SEMANTIC EQUIVALENCE



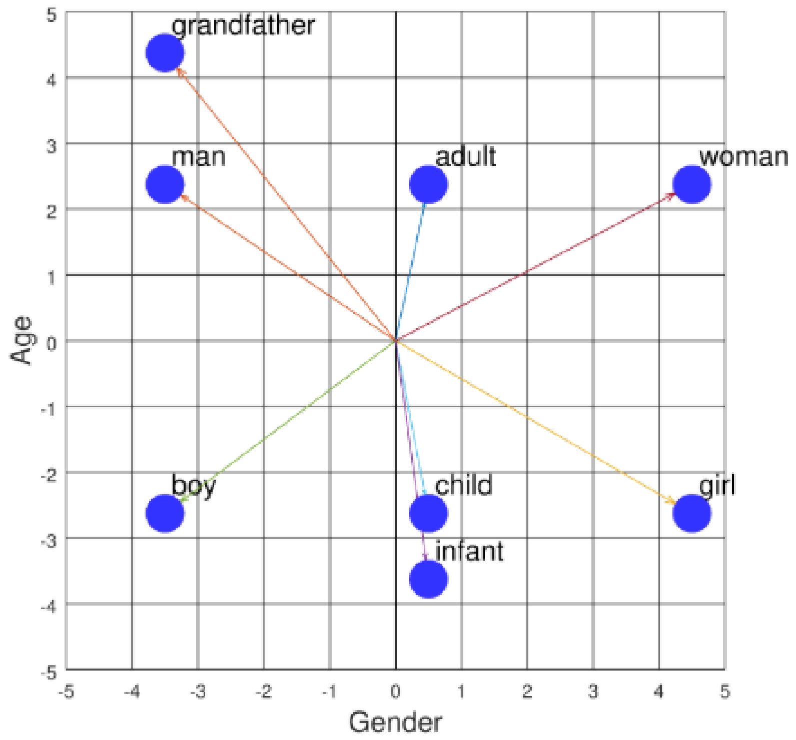
Male-Female



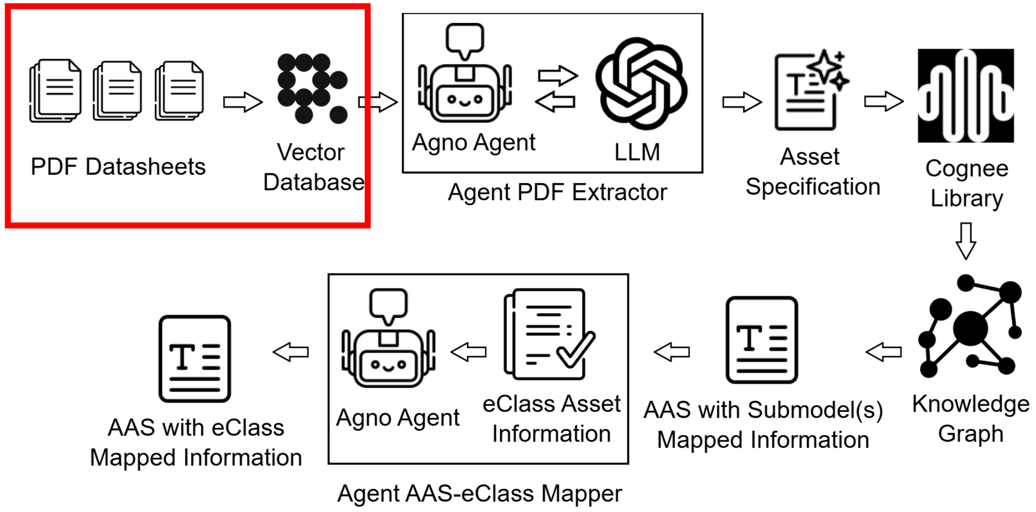
Verb tense



Country-Capital



AGNO FRAMEWORK

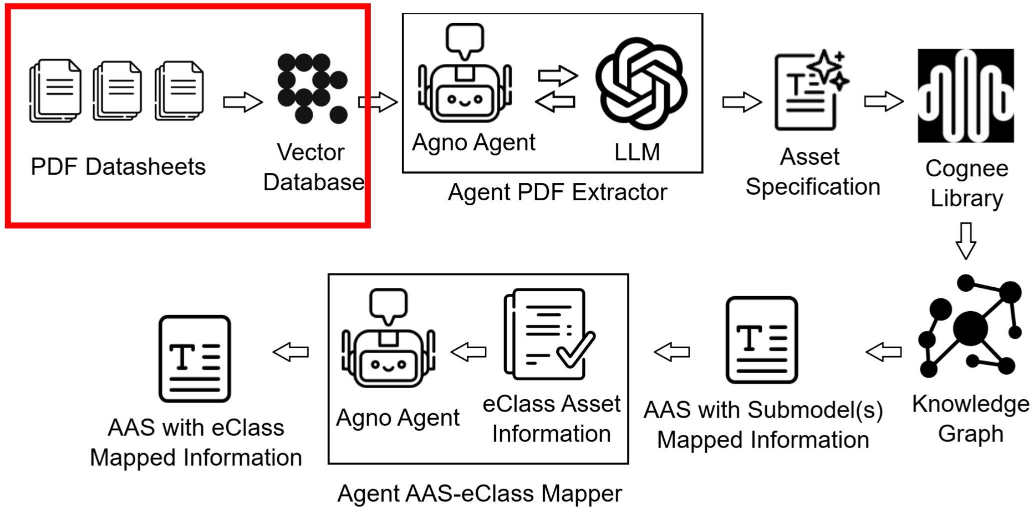


LanceDB

Open-source vector database that can support low-latency billion-scale vector search on a single node.

https://lancedb.com

AGNO FRAMEWORK



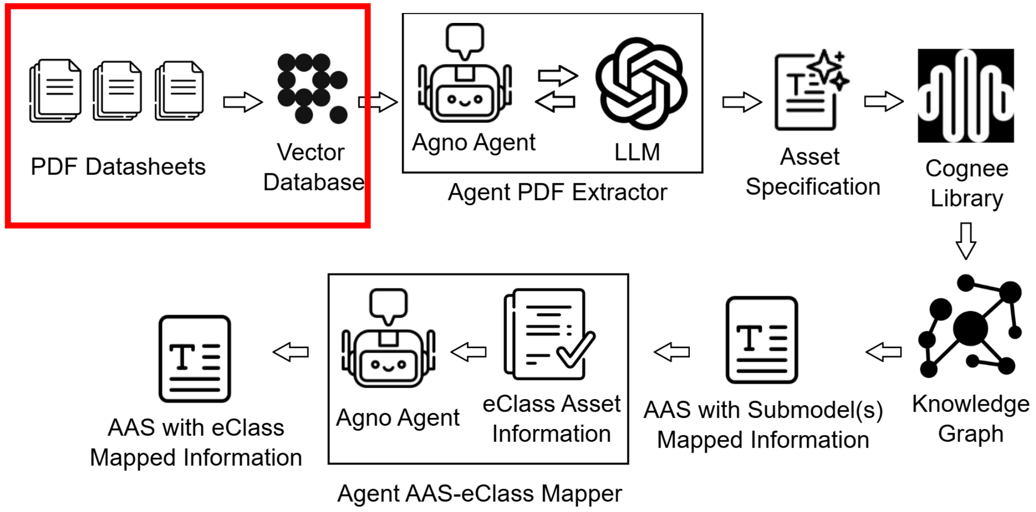
LanceDB

Open-source vector database that can support low-latency billion-scale vector search on a single node.

```
# LanceDB Vector DB
vector_db = LanceDb(
    table_name="recipes",
    uri="/tmp/lancedb",
    search_type=SearchType.keyword,
)

# Knowledge Base
knowledge_base = Knowledge(
    vector_db=vector_db,
)
```

https://lancedb.com

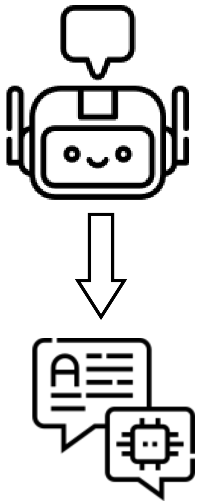
AGNO FRAMEWORK



LanceDB

Open-source vector database that can support low-latency billion-scale vector search on a single node.

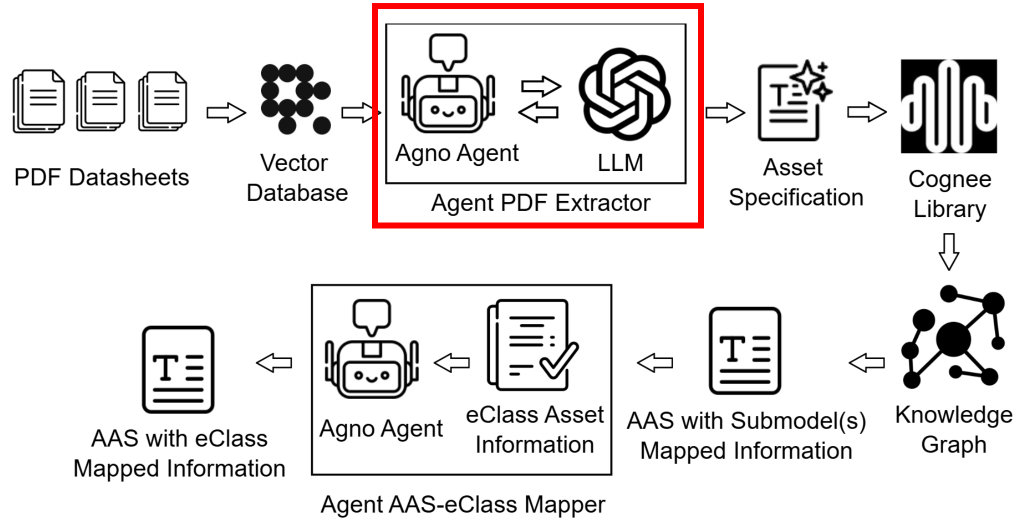
```python
# LanceDB Vector DB
vector_db = LanceDb(
    table_name="recipes",
    uri="/tmp/lancedb",
    search_type=SearchType.keyword,
)

# Knowledge Base
knowledge_base = Knowledge(
    vector_db=vector_db,
)
```



Natural Language Queries

https://lancedb.com

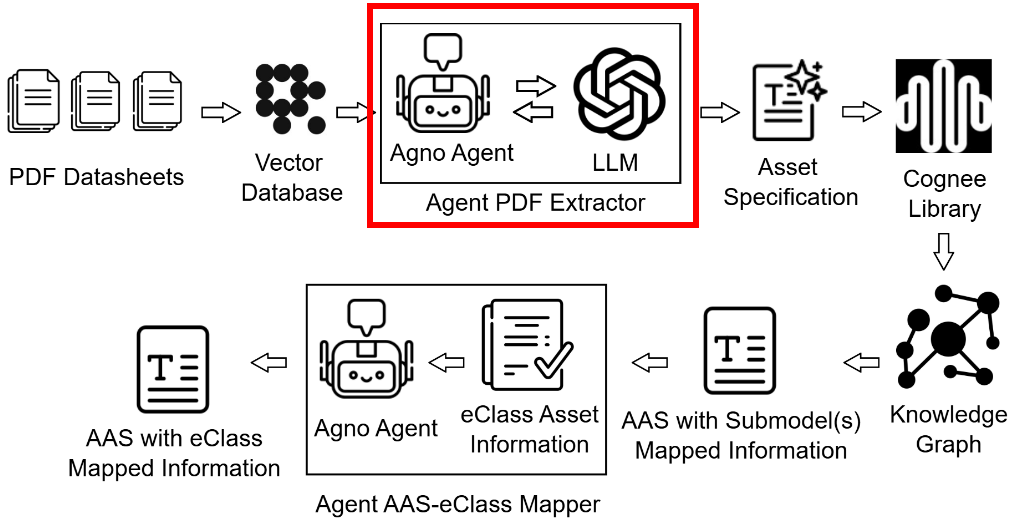
AGNO FRAMEWORK



Agno fast multi-agent framework

https://github.com/agno-agi/agno

AGNO FRAMEWORK



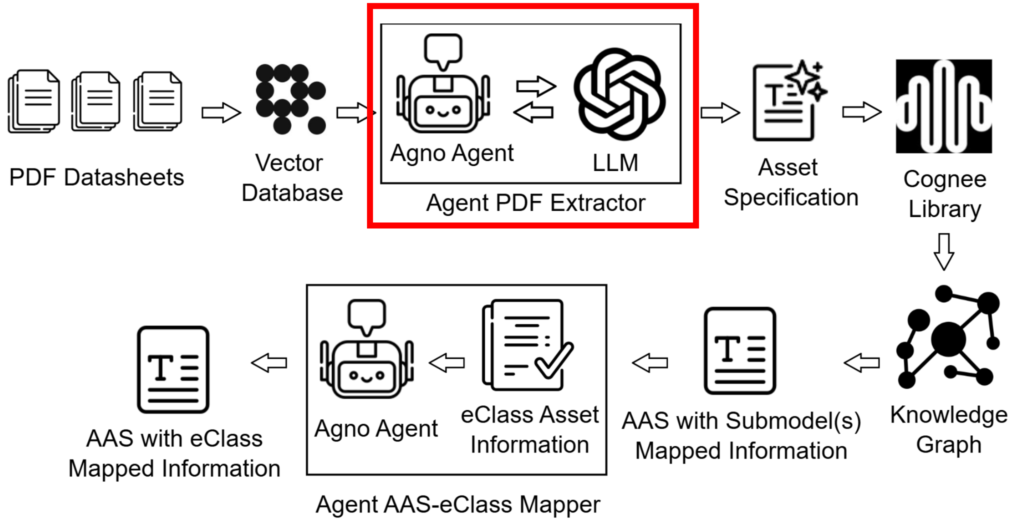
Agno fast multi-agent framework

```python
from agno.agent import Agent
from agno.models.anthropic import Claude
from agno.tools.hackernews import HackerNewsTools

agent = Agent(
    model=Claude(id="claude-sonnet-4-5"),
    tools=[HackerNewsTools()],
    markdown=True,
)
agent.print_response("Write a report on trending startups.", stream=True)
```

You can also specify name, description, instructions, knowledge, etc.

https://github.com/agno-agi/agno

AGNO FRAMEWORK



Agno fast multi-agent framework

```python
from agno.agent import Agent
from agno.models.anthropic import Claude
from agno.tools.hackernews import HackerNewsTools

agent = Agent(
    model=Claude(id="claude-sonnet-4-5"),
    tools=[HackerNewsTools()],
    markdown=True,
)
agent.print_response("Write a report on trending startups.", stream=True)
```

You can also specify name, description, instructions, knowledge, etc.

- ☑ Tool Integration
- ☑ Session Handling
- ☑ Context Management
- ☑ Database Integration

https://github.com/agno-agi/agno

NEW TECHNIQUES, NEW PROBLEMS

NEW TECHNIQUES, NEW PROBLEMS

NEW TECHNIQUES, NEW PROBLEMS, NEW SOLUTIONS

Why Language Models Hallucinate

Adam Tauman Kalai* OpenAI
Ofir Nachum OpenAI
Santosh S. Vempala† Georgia Tech
Edwin Zhang OpenAI

September 4, 2025

Abstract

Like students facing hard exam questions, large language models sometimes guess when uncertain, producing plausible yet incorrect statements instead of admitting uncertainty. Such "hallucinations" persist even in state-of-the-art systems and undermine trust. We argue that language models hallucinate because the training and evaluation procedures reward guessing over acknowledging uncertainty, and we analyze the statistical causes of hallucinations in the modern training pipeline. Hallucinations need not be mysterious—they originate simply as errors in binary classification. If incorrect statements cannot be distinguished from facts, then hallucinations in pretrained language models will arise through natural statistical pressures. We then argue that hallucinations persist due to the way most evaluations are graded—language models are optimized to be good test-takers, and guessing when uncertain improves test performance. This "epidemic" of penalizing uncertain responses can only be addressed through a socio-technical mitigation: modifying the scoring of existing benchmarks that are misaligned but dominate leaderboards, rather than introducing additional hallucination evaluations. This change may steer the field toward more trustworthy AI systems.

1 Introduction

Language models are known to produce overconfident, plausible falsehoods, which diminish their utility and trustworthiness. This error mode is known as "hallucination," though it differs fundamentally from the human perceptual experience. Despite significant progress, hallucinations continue to plague the field, and are still present in the latest models (OpenAI, 2025a). Consider the prompt:

What is Adam Tauman Kalai's birthday? If you know, just respond with DD-MM.

On three separate attempts, a state-of-the-art open-source language model[1] output three incorrect dates: "03-07", "15-06", and "01-01", even though a response was requested only if known. The correct date is in Autumn. Table 1 provides an example of more elaborate hallucinations.

Hallucinations are an important special case of *errors* produced by language models, which we

Kalai, Adam Tauman, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. "Why language models hallucinate." *arXiv preprint arXiv:2509.04664* (2025).

NEW TECHNIQUES, NEW PROBLEMS, NEW SOLUTIONS

Why Language Models Hallucinate

Adam Tauman Kalai* OpenAI
Ofir Nachum OpenAI
Santosh S. Vempala† Georgia Tech
Edwin Zhang OpenAI

September 4, 2025

Abstract

Like students facing hard exam questions, large language models sometimes guess when uncertain, producing plausible yet incorrect statements instead of admitting uncertainty. Such "hallucinations" persist even in state-of-the-art systems and undermine trust. We argue that language models hallucinate because the training and evaluation procedures reward guessing over acknowledging uncertainty, and we analyze the statistical causes of hallucinations in the modern training pipeline. Hallucinations need not be mysterious—they originate simply as errors in binary classification. If incorrect statements cannot be distinguished from facts, then hallucinations in pretrained language models will arise through natural statistical pressures. We then argue that hallucinations persist due to the way most evaluations are graded—language models are optimized to be good test-takers, and guessing when uncertain improves test performance. This "epidemic" of penalizing uncertain responses can only be addressed through a socio-technical mitigation: modifying the scoring of existing benchmarks that are misaligned but dominate leaderboards, rather than introducing additional hallucination evaluations. This change may steer the field toward more trustworthy AI systems.

1 Introduction

Language models are known to produce overconfident, plausible falsehoods, which diminish their utility and trustworthiness. This error mode is known as "hallucination," though it differs fundamentally from the human perceptual experience. Despite significant progress, hallucinations continue to plague the field, and are still present in the latest models (OpenAI, 2025a). Consider the prompt:

What is Adam Tauman Kalai's birthday? If you know, just respond with DD-MM.

On three separate attempts, a state-of-the-art open-source language model[1] output three incorrect dates: "03-07", "15-06", and "01-01", even though a response was requested only if known. The correct date is in Autumn. Table 1 provides an example of more elaborate hallucinations.

Hallucinations are an important special case of *errors* produced by language models, which we

A comprehensive survey on integrating large language models with knowledge-based methods

Wenli Yang [a,*], Lilian Some [a], Michael Bain [b], Byeong Kang [a]

[a] University of Tasmania, Churchill Ave, Hobart, 7005, TAS, Australia
[b] University of New South Wales, High St, Sydney, 2052, NSW, Australia

ARTICLE INFO

Keywords:
LLMs
Knowledge-based
Knowledge integration
RAG
KG

ABSTRACT

The rapid development of artificial intelligence has led to marked progress in the field. One interesting direction for research is whether Large Language Models (LLMs) can be integrated with structured knowledge-based systems. This approach aims to combine the generative language understanding of LLMs and the precise knowledge representation systems by which they are integrated. This article surveys the relationship between LLMs and knowledge bases, looks at how they can be applied in practice, and discusses related technical, operational, and ethical challenges. Utilizing a comprehensive examination of the literature, the study both identifies important issues and assesses existing solutions. It demonstrates the merits of incorporating generative AI into structured knowledge-base systems concerning data contextualization, model accuracy, and utilization of knowledge resources. The findings give a full list of the current situation of research, point out the main gaps, and propose helpful paths to take. These insights contribute to advancing AI technologies and support their practical deployment across various sectors.

1. Introduction

The rapid development of Large Language Models (LLMs) has shaped the contours of AI. This has landed the models in areas where they can excel in generating natural text, never reaching any peak before.LLMs are based on deep learning structures, and can therefore perform effectively in many natural language processing tasks, such as text generation, sentiment analysis, or sophisticated dialogue systems.

Recent surveys have explored diverse aspects of LLMs, such as their architectures, training methodologies, and performance evaluation benchmarks. Many focus on specific topics, such as detailed analyses of state-of-the-art models [1], scaling laws innovations((i.e. what a new law brings to make old predictions still work)) [2], and pretraining methods used with large datasets [3]. Others investigate domain-specific fine-tuning effects from reinforcement learning or human feedback [4,5], and transfer learning strategies [6]. Despite these valuable contributions, however, there remains a paucity of comprehensive perspectives that connect the foundational principles of LLMs with practical applications and the challenges faced as these models are implemented in real-world situations.

This survey addresses this gap by presenting a comprehensive analysis of LLMs' foundational principles and their applications across diverse domains. Although LLMs have achieved remarkable progress, their practical deployment faces challenges. Issues such as interpretability, high computational demands, and scalability impede their broader adoption. This study also investigates the integration of generative AI with knowledge bases, emphasizing how this synergy can mitigate these limitations and unlock new opportunities.

To guide this analysis, several key assumptions are outlined:

Real-World Application Challenges: LLMs encounter substantial obstacles in real-world settings, particularly in terms of interpretability, computational requirements, and scalability, which limit their effectiveness and broader applicability.

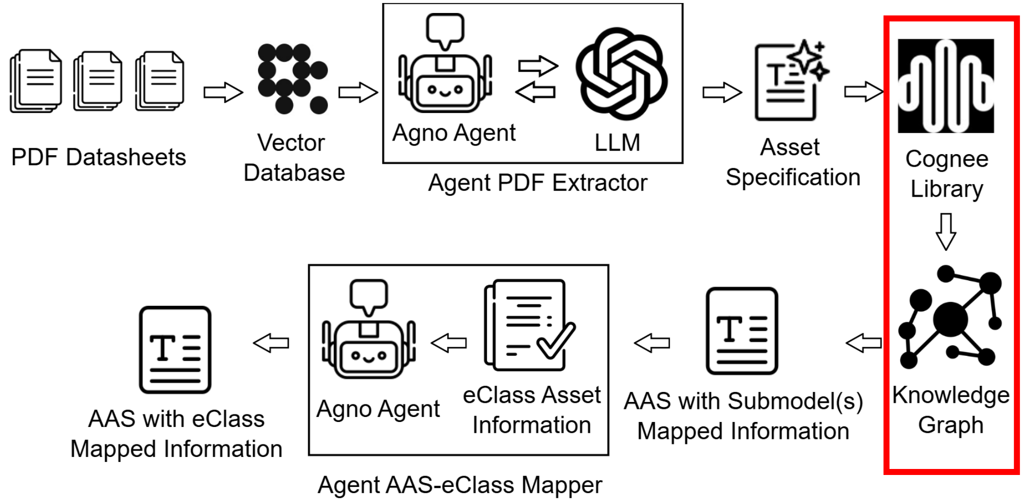
Mitigation through Integration: The integration of LLMs with knowledge bases — using methods such as Retrieval-Augmented Generation (RAG), Knowledge Graphs, and Prompt Engineering — offers promising solutions. This synergy enhances data contextualization, improves model accuracy, and reduces computational costs.

Barriers to Adoption: Persistent challenges, including the need for interpretability, efficient resource utilization, and seamless integration with existing systems, continue to hinder the widespread adoption of LLMs.

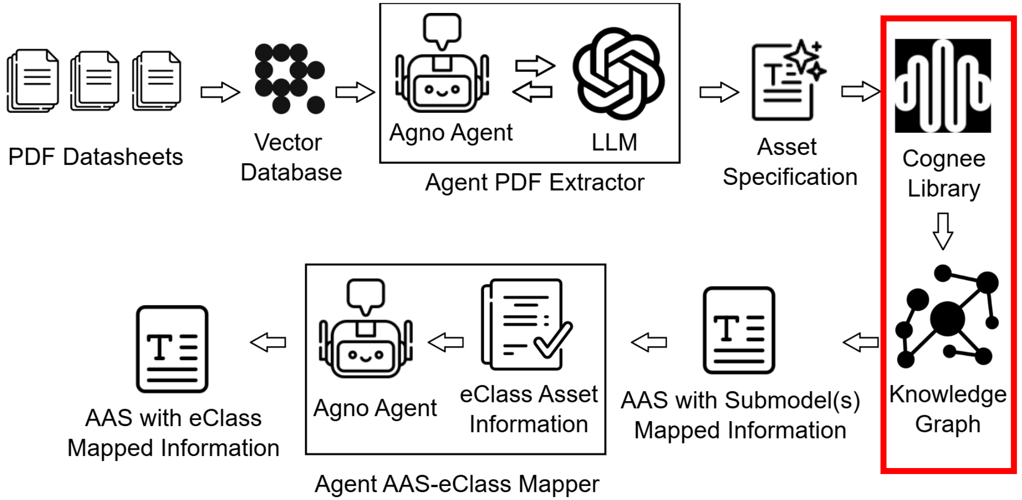
Building on these assumptions, this survey provides a structured and integrated analysis of LLMs. The primary contributions are as follows:

Kalai, Adam Tauman, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. "Why language models hallucinate." *arXiv preprint arXiv:2509.04664* (2025).
Yang, Wenli, Lilian Some, Michael Bain, and Byeong Kang. "A comprehensive survey on integrating large language models with knowledge-based methods." *Knowledge-Based Systems* (2025).

NEW TECHNIQUES, NEW PROBLEMS, NEW SOLUTIONS



NEW TECHNIQUES, NEW PROBLEMS, NEW SOLUTIONS



cognee

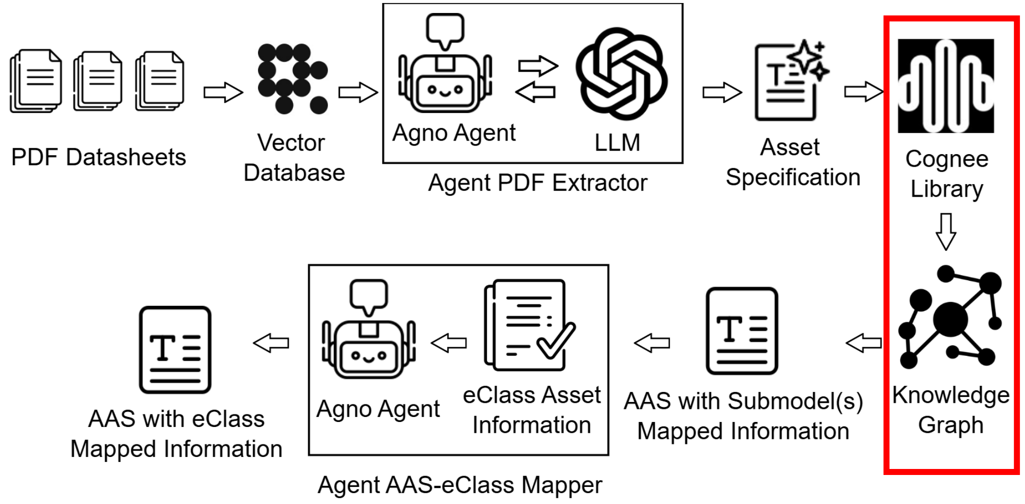
Cognee turns your data into a knowledge graph.

`.add`

Send in your data asynchronously.

https://github.com/topoteretes/cognee

NEW TECHNIQUES, NEW PROBLEMS, NEW SOLUTIONS



cognee

Cognee turns your data into a knowledge graph.

`.add`

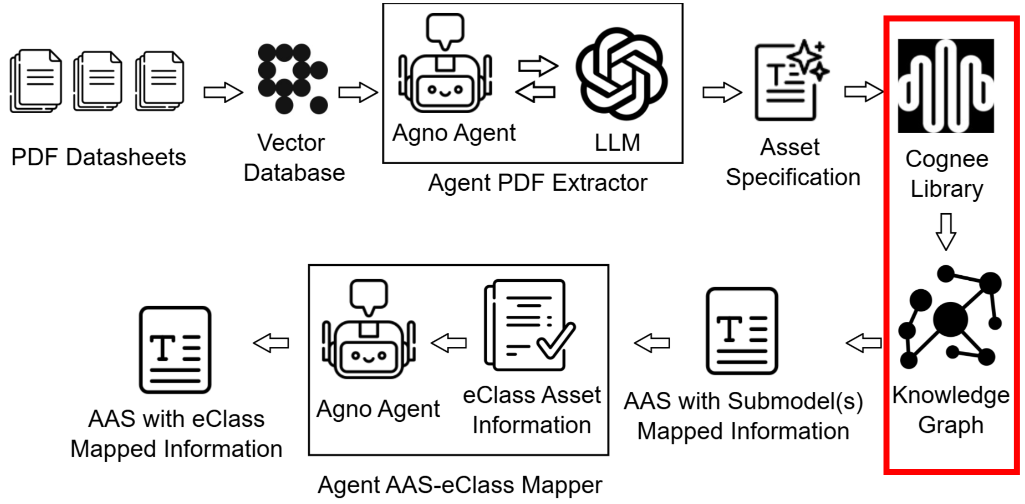
Send in your data asynchronously.

`.cognify`

Cognee splits your documents into chunks, extract entities, relations, and links it all into a queryable graph.

https://github.com/topoteretes/cognee

NEW TECHNIQUES, NEW PROBLEMS, NEW SOLUTIONS



cognee

Cognee turns your data into a knowledge graph.

`.add`

Send in your data asynchronously.

`.cognify`

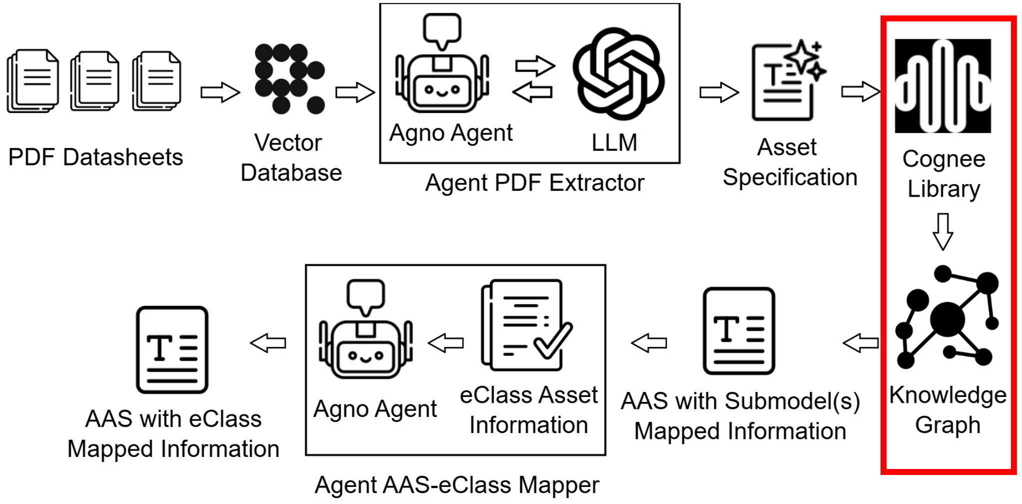
Cognee splits your documents into chunks, extract entities, relations, and links it all into a queryable graph.

`.search`

Queries combine vector similarity with graph traversal.

https://github.com/topoteretes/cognee

NEW TECHNIQUES, NEW PROBLEMS, NEW SOLUTIONS



cognee

Cognee turns your data into a knowledge graph.

`.add`

Send in your data asynchronously.

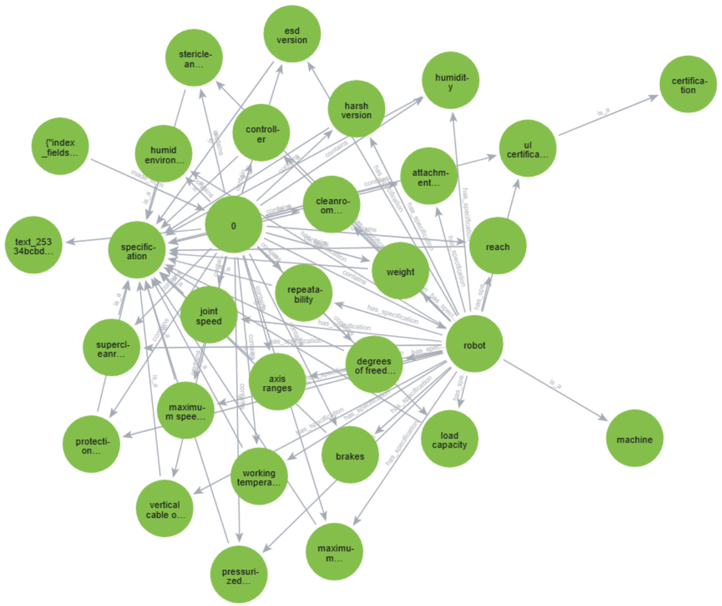
`.cognify`

Cognee splits your documents into chunks, extract entities, relations, and links it all into a queryable graph.

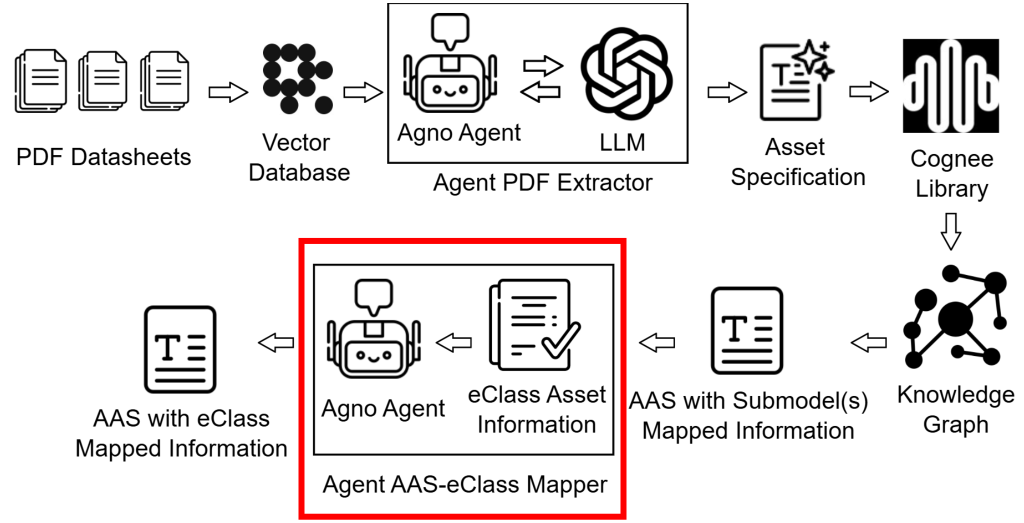
`.search`

Queries combine vector similarity with graph traversal.

https://github.com/topoteretes/cognee



CUSTOM INFORMATION MAPPING



Asset Mapped Information AAS fields

eClass Standard for Mapped Fields

Enriched AASX files

https://eclass.eu/en

CONCLUSION

CONCLUSION

Digital assets and AAS are fundamental for Industry 4.0/5.0.

CONCLUSION

Digital assets and AAS are fundamental for Industry 4.0/5.0.

AI Agents and LLMs help automating the AAS adoption process.

CONCLUSION

Digital assets and AAS are fundamental for Industry 4.0/5.0.

AI Agents and LLMs help automating the AAS adoption process.

AI Agents and Knowledge Graphs represent a powerful combination.

ACCELERATING THE ADOPTION OF ASSET ADMINISTRATION SHELLS THROUGH AI AGENTS

DR. CAMILO VELÁZQUEZ RODRÍGUEZ (camilo.velazquez@flandersmake.be)

STEVEN KAUFFMANN

TOM PAUWAERT

STIJN HUYSENTRUYT

AXL VAN ALBOOM

The Fifteenth International Conference on Intelligent Systems and Applications (INTELLI) 2026

March 08 - 12, 2026

Valencia, Spain