Rethinking Software & Soft. Eng. in the Foundation Model Era (FM)

A Curated Catalogue of Challenges

Ahmed E. Hassan, IEEE/ACM/Steacie Fellow, Mustafa Laureate, Queen's University, Canada
Ahmed@cs.queensu.ca

Joint effort with Dayi Lin, Gopi Krishnan Rajbahadur, Keheliya Gallaba, Filipe R. Cogo, Boyuan Chen, Haoxiang Zhang, Kishanthan Thangarajah, Gustavo Ansaldi Oliva, Jiahuei Lin, Wali Mohammad Abdullah, and Zhen Ming Jiang

FMArts Dev team (alphabetically by last name): Jack Basha, Aaditya Bhatia, Charles Chang, Ximing Dong, Azmain Kabir, Hao Li, Yu Shi, Cedric Wang, Shaowei Wang, Xu Yang, and Sky Zhang.

How to cite this session?

```
@misc{Hassan2024AIwareTutorialChallenges,

author = {Hassan, Ahmed E. and Lin, Dayi and Rajbahadur, Gopi Krishnan and Gallaba, Keheliya and Cogo, Filipe Roseiro and Chen, Boyuan and Zhang, Haoxiang and Thangarajah, Kishanthan and Oliva, Gustavo and Lin, Jiahuei (Justina) and Abdullah, Wali Mohammad and Jiang, Zhen Ming (Jack)},

title = {Rethinking Software and Software Engineering in the Foundation Model Era},

howpublished = {Tutorial presented at the AIware Leadership Bootcamp 2024},

month = {November},

year = {2024},

address = {Toronto, Canada},

note = {Part of the AIware Leadership Bootcamp series.},

url = {https://aiwarebootcamp.io/slides/2024_aiwarebootcamp_hassan_rethinking_software_and_se_in_the_fm_era.pdf}}
```

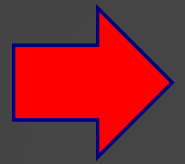
Check this paper for more information about this session

```
@inproceedings{Hassan2024FSE,

author = {Hassan, Ahmed E. and Lin, Dayi and Rajbahadur, Gopi Krishnan and Gallaba, Keheliya and Cogo, Filipe Roseiro and Chen, Boyuan and Zhang, Haoxiang and Thangarajah, Kishanthan and Oliva, Gustavo and Lin, Jiahuei (Justina) and Abdullah, Wali Mohammad and Jiang, Zhen Ming (Jack)},

title = {Rethinking Software Engineering in the Era of Foundation Models: A Curated Catalogue of Challenges in the Development of Trustworthy FMware},

year = {2024},

publisher = {Association for Computing Machinery},

address = {New York, NY, USA},

url = {https://doi.org/10.1145/3663529.3663849},

booktitle = {Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering},

pages = {294–305},

numpages = {12},

series = {FSE 2024}

}
```

Rain-Sensing Windshield Wipers

Codeware

Optical/Infrared/Acoustic
Sensors
+
Lots source code



AIware

ML/FM + a camera
+ lots of "tagged" pictures
+
"ZERO" source code

Developer writes code

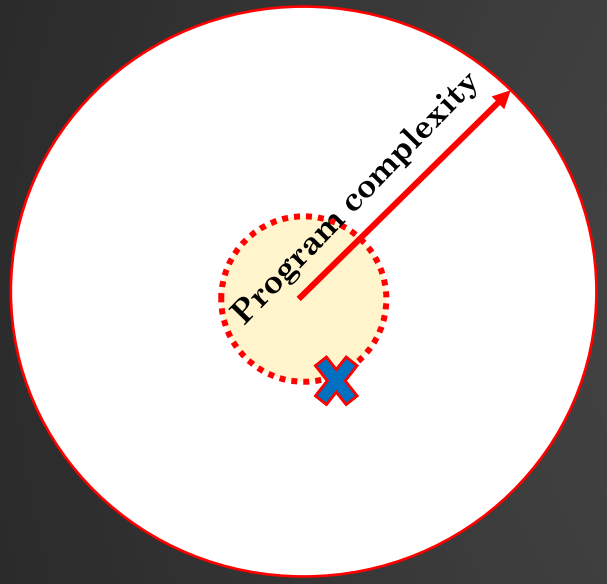
Developer defines a *search space* with *data* as the new code

Fix a bug: "change code"

Fix a bug: "take more pics"!!

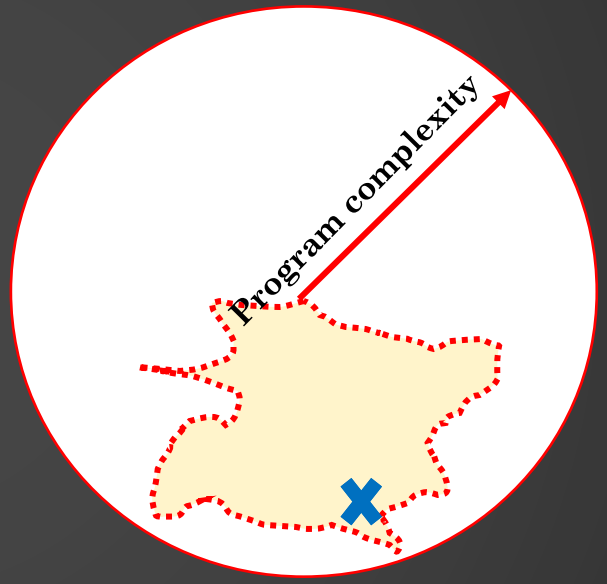
Code Programming by SW-Expert
➔ Alignment Programming by SW-Makers

Codeware



Software complexity is limited by human's ability to express a complex problem

AIware



Software complexity is limited by the "quality" of the tagged data as AI searches for the "software solution" that maximizes a fitness function

Foundation Models are Software's Calculator Moment!

- Millions of Software Developers
  ➔ Billions of Software Makers!!!
- The Happy Developer!
- 10X Developer!

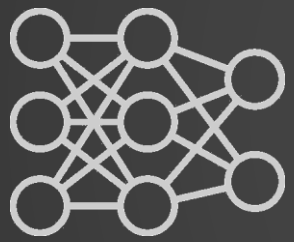
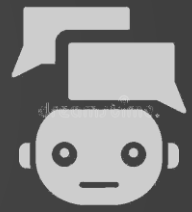
28.7 Million Developers versus a world population of 8 Billion.

FMs enable more complex AIware; more/closer software maker (end-user & domain-expert) involvement, but need a great amount of engineering to happen

The Evolution of Software Generations

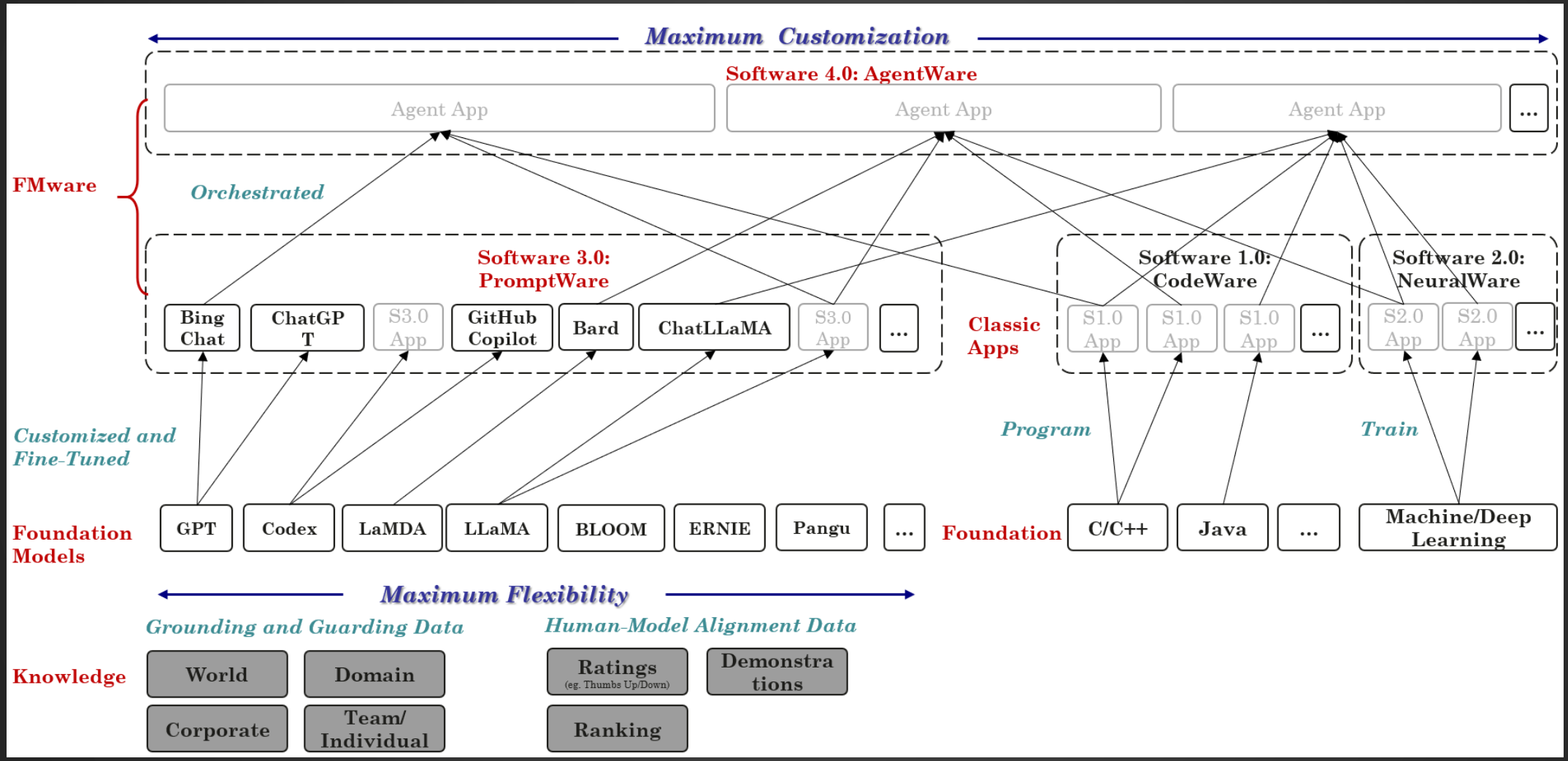
each with a new form, lifecycle, managed assets, and roles
(aka Engineering Paradigm)

FMware

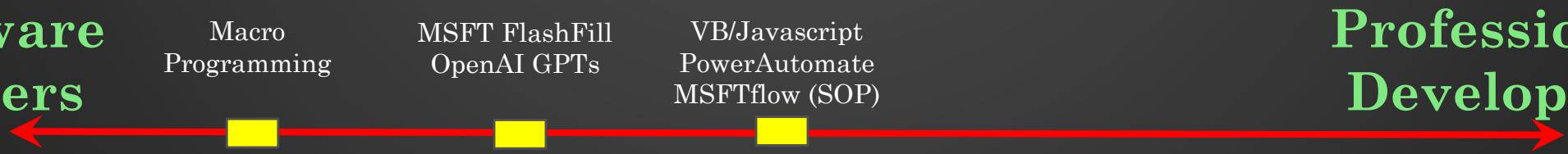
Codeware	Neuralware	Promptware	Agentware	Mindware
				
Programmer express logic by writing code	*ML Engineer express logic by training model with data*	*Software makers express logic through foundation model prompts and workflows*	*Agents making decisions, taking actions, and interacting with other agents*	*AGI with human defined constitutions and oversight*

AIware

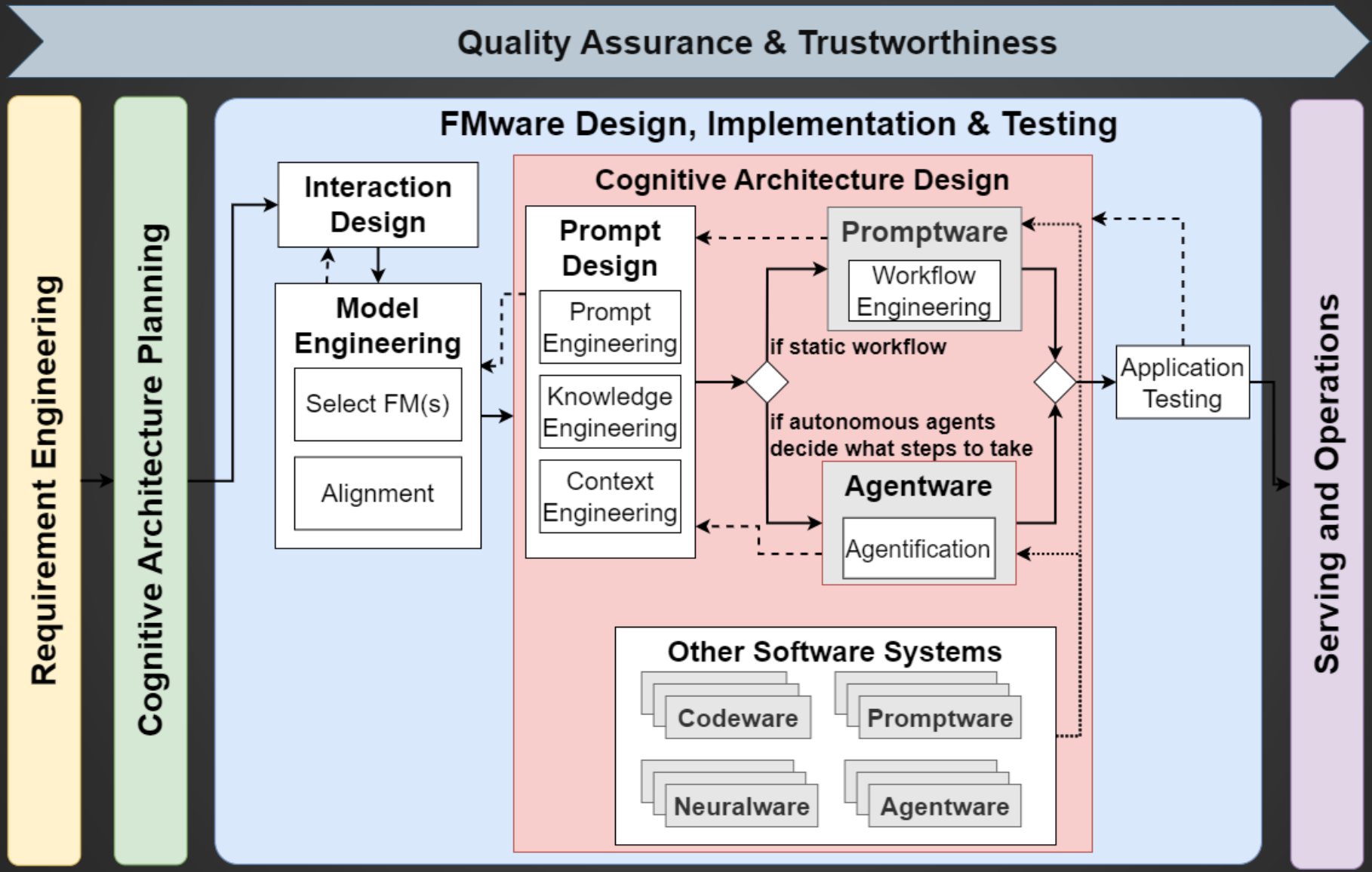
Future Software is a Composition of Many Generations of Software and FMs



AIware enables Software Makers & Developers to Create Different and Richer Types of Software



The FMware lifecycle



Productionizing AI/FMware is HARD!!

"There is a large class of problems that are easy to imagine and build demos for, but extremely hardto make products out of. For example, self-driving:

It's easy to demo a car self-driving around a block, but making it into a product takes a decade." Karpathy

Productionizing AI/FMware is HARD!!

LinkedIn Engineering Blog

Consistent quality… The team achieved 80% of the basic experience we were aiming to provide within the first month and then spent an additional four months attempting to surpass 95% completion of our full experience - …., the initial pace created a false sense of 'almost there,' which became **discouraging as the rate of improvement slowed significantly for each subsequent 1% gain.**

"most of these, like each of these tests, would **probably cost 1-2 cents to run,** but once you end up with a lot of them, that will start adding up anyway". P4 attempted to automate testing but was asked to stop their efforts because of costs in running benchmarks, and instead would only run a small set of them manually after large changes

GitHub Microsoft

" The complexity of these systems surpasses anything that we have seen before, Neuralware was hard this stuff is REALLY HARD!! Very few people trained and they are too pricey!!"

Red Hat

Shipping FMware is HARD!!

Challenges in productionizing FMware

Intrinsic limitations of FMs

Complex Tasks Limitations

Hallucination Limitations

Closed Loop Limitations

Engineering pain points

Low Productivity

High Risks

High Costs

Managing alignment data

Design, Dev. & Test | Productivity | Operational Cost

Problems

- **New data types, slow and costly to curate** (instructions, reference response in NL, ratings & rankings)
- **Data leakage** from evaluation
- Accurately and quantitatively **gauge adequate amount** of high-quality alignment data

State of practice

- **Active learning & weak/self supervision**: Snorkel (hard to debug), Self-Instruct (model collapse)

Innovation path

- **Low-manual-intervention auto labelling**
- **Open and inner source data, and its license compliance**

Crafting effective prompts

Design, Dev. & Test | Productivity

Problems

- **Prompts are too low level and fragile**, sensitive to tiny changes, requiring trial-and-error
- **Prompts are non-transferrable** and cannot co-evolve with models or be ported across models
- **Lack of transparency and reproducible results,** hard to debug

State of practice

- **Inference transparency** (sampling probability, token-level attribution), no higher level abstraction for developers
- **Automated prompt optimization,** mostly at token level in isolation from other factors (e.g., models, parameters, hardware environment)

Innovation path

- **Higher level prompting and transparency**
- **Reproducible prompting**
- **Going beyond prompt hacking to intent programming**

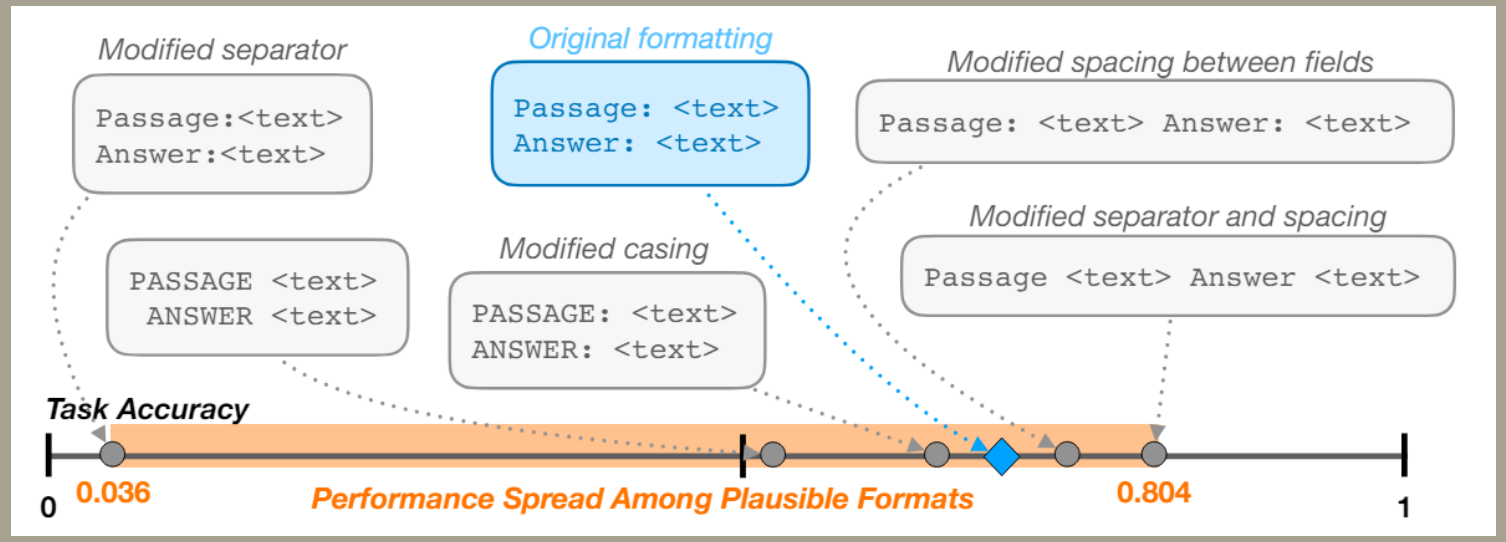
Crafting effective prompts

Design, Dev. & Test | Productivity

Problen *ion path*

- **Prompts a** **level and** sensitive t requiring t
- **Prompts a** **transferra** co-evolve be ported
- **Lack of transparency and reproducible results,** hard to debug from other factors (e.g., models, parameters, hardware environment)

el **and** **ıcy** **ble** **ond** **ıcking to** **gramming**



"How I learned to start worrying about prompt formatting", Sclar et al.

Multi-generation software

Design, Dev. & Test | Productivity

Problems

- **Integrating different generations of legacy software**, with different data models and communication protocols
- **Changes in the shared legacy software systems impact downstream FMware**

State of practice

- **Plug-ins** enabled by users for promptware
- **Packaging legacy software as tools that FMware can reach**. Either requires developer to repackage existing tools to be compatible

Innovation path

- **Recognizing that FMware is rarely a green-field project**
- **Reusable, modular integration of legacy software and *data***

Degree of controllability

Design, Dev. & Test | Serving & Ops. | Risk

Problems

- **Software in certain domains require robust controllability**, but FMware is inherently stochastic

State of practice

- **Restricted approaches** such as Standard Operation Procedures (SOP) and Standard Work Instructions (SWI) in prompts or cognitive architecture
- **Human oversight:** communicating plans with human before execution

Innovation path

- **Fine-grained boundaries** (e.g., application permission model in Android)
- **Balancing high-level controllability and low-level autonomy of cognitive architectures**

Compliance & regulations

Requirements Eng. | Design, Dev. & Test | Serving & Ops. | Risk

Problems

- Government regulations such as EU AI Act requires **transparency and human oversight**
- GDPR requires personal data be protected but **using third-party FM service requires data transmission**
- **License compliance for data usage and OSS models**

State of practice

- **Manual and process heavy efforts** such as documentation and risk management framework
- **Data and AI licenses** that focus on individual FM or dataset, disregarding license interaction and agent evolution

Innovation path

- **Tools capable of automatically determine FMwareBOMs**
- **Testing and formal verification of regulatory and license compliance**
- **Privacy-preserving tech for FMware data transfer**

Limited collaboration support

Design, Dev. & Test | Productivity

Problems

- **What to version and how,** given new asset types such as (structured) prompts and (evolving) agents
- **What formats and standards to use.** Lack of standards severely hamper interoperability
- **How to package and distribute** e.g. evolving agents and its data

State of practice

- **Treat all assets as text and record in Git**, ignoring the uniqueness of new assets and cannot facilitate efficient collaborative development or reuse
- **No established standards, no established hub**

Innovation path

- **Granular version control to facilitate collaborative development and reuse**
- **Standards and protocols for interoperability**

Operation & semantic observability

Serving & Ops. | Productivity | Risk | Operational Cost

Problems

- **Need for enhanced operation telemetry,** monitoring token consumptions, groundedness, environment interaction etc.
- **Need for semantic signal telemetry** to monitor explicit and implicit user feedback

State of practice

- Several closed-source platform provides **observability into model inference and component interactions**, not at dynamic multi-agent level
- **Semantic signal is used by GitHub Copilot** , no general solution available

Innovation path

- **Observe diverse aspects of enhanced metrics of complex FMware**
- **Advance generalizable semantic signal telemetry**

Challenge 7

Performance engineering

Serving & Ops. | Productivity | Operational Cost

Problems

- Performance of FMware drops with **longer prompts and multi-round inference.**
- FMware may involve multiple FMs dynamically. **FM level SLA does not guarantee FMware level SLA.**
- Self-hosted FMware may need to **load and offload FMs** frequently.

State of practice

- **Optimizing for single-round inference**
- **Using / creating smaller FMs**
- **Compress prompts**
- **Semantic caching**

Innovation path

- **Intent-preserving declarative abstraction that can be optimized at FMware level**
- **FMware level SLA guarantee**
- **Impact of complex cognitive architectures on latency**

Testing under non-determinism

Design, Dev. & Test | Productivity | Risk

Problems

- **Generative tasks has no ground truth.**
- **Every test is flaky. Test results are not reproducible.**
- Both input space and output space become **impossible to exhaust. Cannot specify test cases.**
- **High cost** to run test cases.

State of practice

- **Human evaluation** , but slow and costly.
- **FM as an evaluator (e.g., GPT-4)** , but its correlation with human evaluation is uncertain and can be low. FM also cannot accurately provide scaled answers.

Innovation path

- **Define and evaluate quality of FMware with low human effort and high reliability.**

Siloed tooling & lack of process

All phases (cross-cutting) | Productivity | Risk | Operational Cost

Problems

- **Explosion of siloed tools and solutions**, causing cognitive overload, context switching, and need for glue code.
- Lack of standardized development tools and practices lead to **non-automated and inefficient compliance & governance.**

State of practice

- **Ongoing efforts by key industry players** (e.g., PromptFlow by Microsoft, LangSmith by LangChain). However they either do not cover full lifecycle of FMware, missing key considerations (e.g., compliance), or offer little extensibility.

Innovation path

- **Unified platform that provides cradle-to-grave lifecycle support and extensibility.**

FMArts: Our FMware lifecycle engineering stack

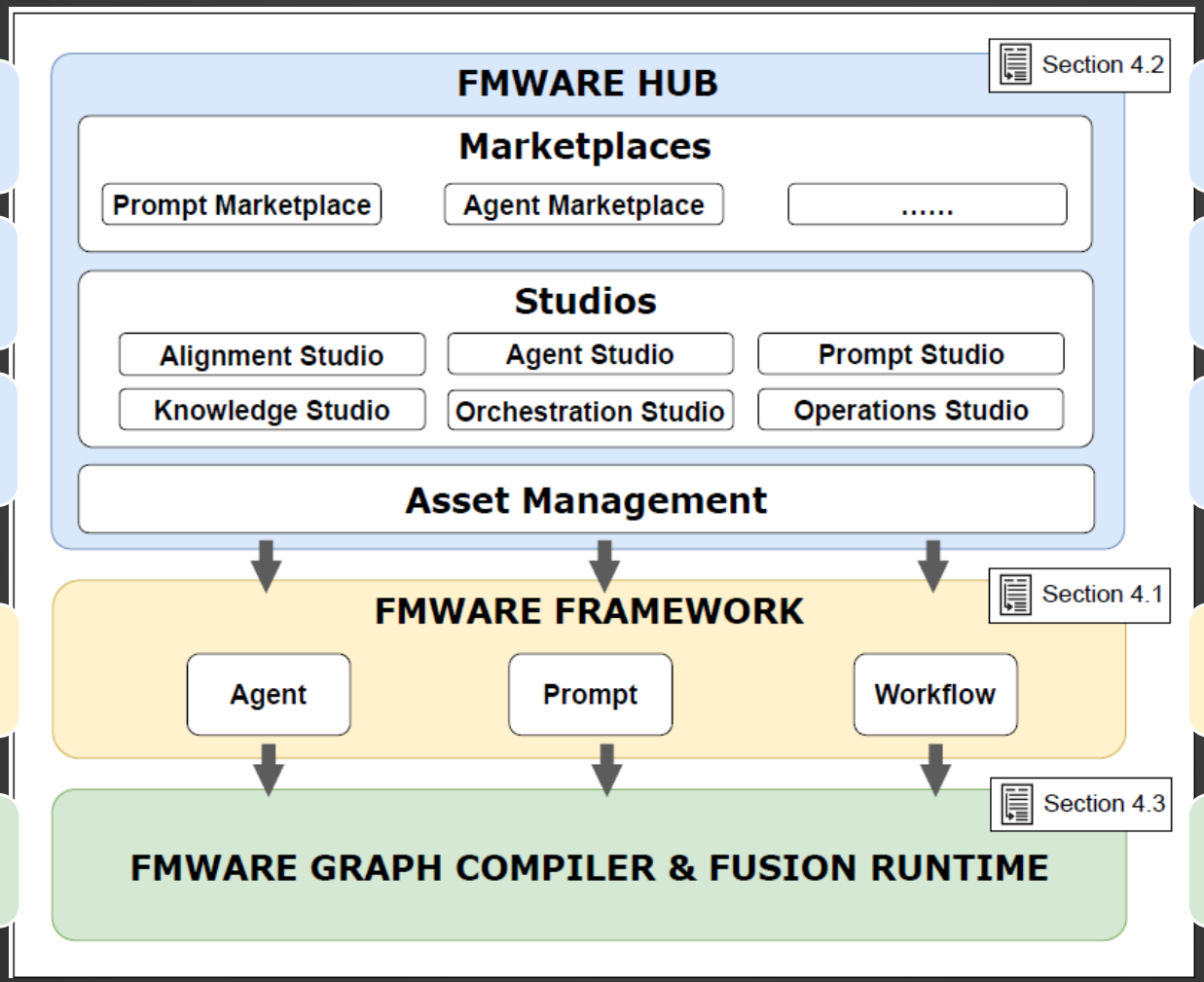
C1: Managing alignment data

C5: Compliance & regulations

C9: Testing under non-determinism

C3: Multi-generation software

C7: Operation & semantic observability



C2: Crafting effective prompt

C6: Limited collaboration support

C10: Siloed tooling & lack of process

C4: Degree of controllability

C8: Performance engineering

We are just starting… Tackling these challenges requires technology breakthroughs, standards, and community efforts

- **C1: Managing alignment data**
- **C2: Crafting effective prompt**
- **C3: Multi-generation software**
- **C4: Degree of controllability**
- **C5: Compliance & regulations**
- **C6: Limited collaboration support**
- **C7: Operation & semantic observability**
- **C8: Performance engineering**
- **C9: Testing under non-determinism**
- **C10: Siloed tooling & lack of process**

More details at:
https://arxiv.org/abs/2402.15943

https://aiwareconf.org
https://fmse.io/
https://opea.dev/
https://se4ai.org/

We are just starting… Tackling these challenges requires technology breakthroughs, standards, and community efforts

C1: Managing alignment data

C5: Compliance & regulations

https://se4ai.org/

1st ACM International Conference on AI-powered Software (AIware)

AIware

Co-located with FSE'24 | July 15-16, 2024 | Porto de Galinhas, Brazil

Not your regular conference!

Main Track	Challenge Track	Industry Statements and Demo Track	Late breaking Arxiv Track
Short presentations and more discussions!		*Only need to fill a google form!*	*No peer-review! Just Arxiv!*

LLM4Code

LLM4Code 2025

The Second International Workshop on Large Language Models for Code
Co-Located with ICSE 2025
Ottawa, Ontario, Canada
May 3, 2025
Follow us on Twitter: @llm4code!
Past Workshops: 2024

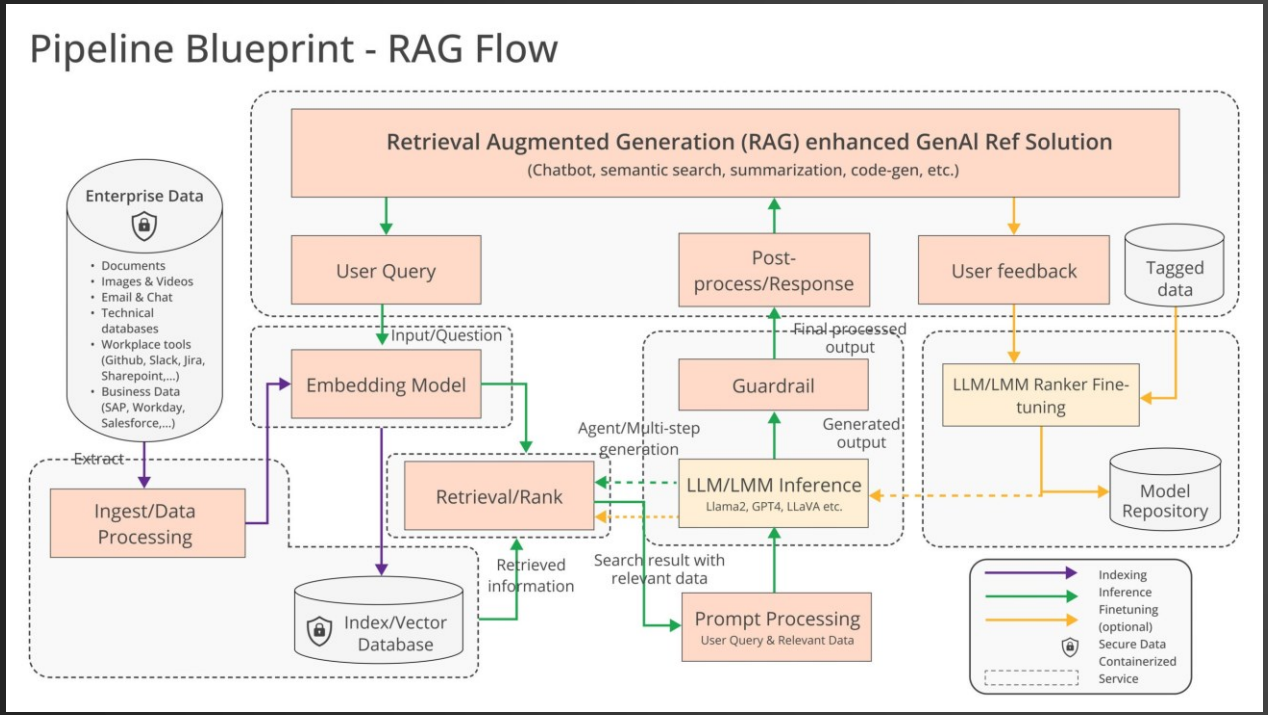
Home Program Speakers Call for Papers Important Dates Organization Program Committee

News

- Oct, 2024: Our HotCRP site is ready for your submissions! Deadline: Monday Nov 18, 2024, 11:59:59 PM AoE.

About

OPEA


Pipeline Blueprint - RAG Flow
Retrieval Augmented Generation (RAG) enhanced GenAI Ref Solution
(Chatbot, semantic search, summarization, code-gen, etc.)
Enterprise Data
Documents
Images & Videos
Email & Chat
Technical databases
Workplace tools (Github, Slack, Jira, Sharepoint,...)
Business Data (SAP, Workday, Salesforce,...)
User Query
Post-process/Response
User feedback
Tagged data
Input/Question
Embedding Model
Guardrail
Final processed output
LLM/LMM Ranker Fine-tuning
Agent/Multi-step generation
Generated output
Extract
Ingest/Data Processing
Retrieval/Rank
LLM/LMM Inference
Llama2, GPT4, LLaVA etc.
Model Repository
Retrieved information
Search result with relevant data
Index/Vector Database
Prompt Processing
User Query & Relevant Data
Indexing
Inference
Finetuning (optional)
Secure Data
Containerized Service


aiven
AMD
anyscale
Articul8
BONC 东方国信
ByteDance
Canonical
CDW
China unicom中国联通
clarifai
CLOUDERA
Corsha
DATASTAX
Datastrato
docker
Domino
dstack
FRONTIERX
Haystack by deepset
Hugging Face
Infosys Navigate your next
intel.
Iterate.ai
JFrog
KX
LlamaIndex
MariaDB Foundation
MINIO
MongoDB.
neo4j
OpenEuler
pathway
prediction Guard
drant
Red Hat
Redis
Rivos
SAP
sas
TURINTECH
vectara
vmware by Broadcom
wipro
Yellowbrick
zensar

SPDX


SPDX 3.0 Revolutionizes Software Management in Systems with Enhanced Functionality and Streamlined Use Cases
THE LINUX FOUNDATION | 16 APRIL 2024
THE LINUX FOUNDATION
SPDX®3.0

LF AI & Data Blog

Introducing the Model Openness Framework: Promoting Completeness and Openness for Reproducibility, Transparency and Usability in AI

By cakerly | April 17, 2024 | No Comments

Hassan et al., AIware Leadership Bootcamp, Toronto, Canada, 2024

MSR 2025


MSR 2025
ttawa
MSR 2025
22nd International Conference on
Mining Software Repositories
April 28-29, Ottawa, Canada

We are just starting… Tackling these challenges requires technology breakthroughs, standards, and community efforts

- **C1: Managing alignment data**
- **C2: Crafting effective prompt**
- **C3: Multi-generation software**
- **C4: Degree of controllability**
- **C5: Compliance & regulations**
- **C6: Limited collaboration support**
- **C7: Operation & semantic observability**
- **C8: Performance engineering**
- **C9: Testing under non-determinism**
- **C10: Siloed tooling & lack of process**

More details at:
https://arxiv.org/abs/2402.15943

https://aiwareconf.org
https://fmse.io/
https://opea.dev/
https://se4ai.org/