Big Data execution in the age of AI

From scientific-scale data systems to Industry 4.0 decision loops

| Stefano Colafranceschi - James Madison University, CERN, Sharpsat |

A brief introduction

- Faculty at **James Madison University**, Integrated Science and Technology with focus on energy, manufacturing and computing.
- Applied Physicist with the **Compact Muon Solenoid (CMS)** experiment at **CERN**, where large-scale data and analysis are part of the scientific workflow.
- Chief Technical Officer/founder **Sharpsat** a remote sensing startup backed by European Space Agency
- Software Developer for **educational tools, open-source data-analysis, and outreach** - including webcasts, media, and technical events.

Faculty at JMU
Systems Thinking, Holistic Engineering

CMS at CERN
Scientific nad large-scale data-analysis

Sharpsat
Satellite Tech and AI modelling

Software and outreach
Education, Analysis, Webcast, Media

Across research, teaching, software, and startups, the same question keeps returning:
How do we turn large data flows into trusted action?

What this talk will do

1. Define what **execution** means in big data systems.
2. Walk through the **generational evolution** of big data.
3. Translate lessons from **science, satellites, software, and outreach** into Industry 4.0.
4. End with a practical architecture and maturity path for **AI-native industrial operations**.

not a Hadoop or similar history lesson

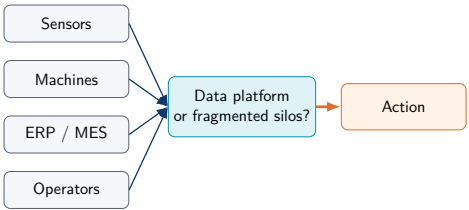
not a (very) technical discussion

not an AI hype pitch

..rather a decision systems talk..

Industry 4.0 has a data problem before it has an AI problem

- Factories, supply chains, labs, utilities, and fleets now produce **continuous machine data**.
- Most organizations still run on **fragmented pipelines**: **Programmable Logic Controller (PLC)** data here, **Enterprise Resource Planning (ERP)** there, quality records elsewhere.
- AI does not remove this complexity. It amplifies it.



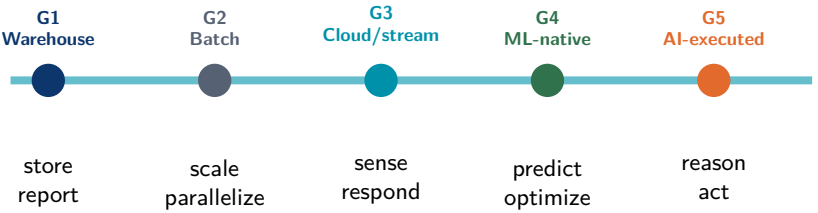
So the new question is: **Can your data system execute fast enough, safely enough, and contextually enough to support action?**

A useful definition: big data execution

Execution is the ability to move from signal to decision to intervention, within the time, cost, and trust constraints of the system.

Speed	Fidelity	Closure
Latency that matches the process: milliseconds, minutes, or weeks.	Enough quality, lineage, and context to trust the output.	The system can trigger human or machine action, not just display a dashboard.

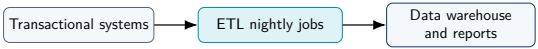
The generational evolution of big data execution



These generations are execution paradigms, not adoption levels. The dominant constraint moved from capacity → scalability → latency → intelligence → autonomy.

Generation 1: the warehouse era

- Core idea: collect structured records into a central warehouse.
- Primary value: reporting, auditability, finance, enterprise visibility.
- Execution model: **batch Extract, Transform, Load (ETL)**, overnight refresh, monthly optimization cycles.
- Limitation for operations: by the time the answer arrives, the process has already moved on.



Industrial analogy

A weekly production review is useful. It is not a control loop.

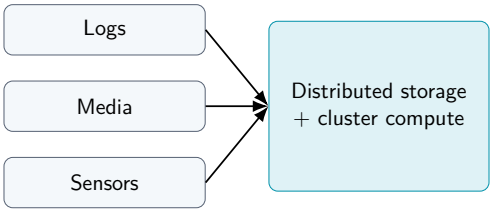
What Generation 1 got right

- **Standardization**: one definition of product, asset, customer, order.
- **Governance**: data quality and reconciliation became serious disciplines.
- **Institutional memory**: organizations could compare performance across time.
- **Lesson for AI age**: if semantics are broken, scaling compute only scales confusion.

The first victory of big data was consistency, not speed

Generation 2: distributed batch at web scale

- Explosion of logs, images, clickstreams, documents, and sensor archives.
- Compute moved to the data with distributed file systems and cluster execution.
- The mental model became: **store everything, parallelize later**.
- This era normalized data lakes, commodity clusters, and large-scale retrospective analytics.



Generation 2 lesson: scale solves some problems and creates others

Wins

- cheaper storage
- broader data coverage
- large offline training sets
- reproducible batch analytics

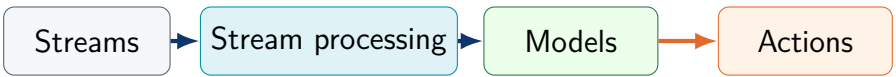
New pain

- lake without semantics
- many copies of the same truth
- slow movement from insight to action
- data engineering complexity explodes

At scale, the bottleneck moves from storage to orchestration.

Generation 3: cloud, streaming, and event-driven operations

- Systems started to care about **time of arrival**, not just time of analysis.
- Streaming pipelines, message buses, and cloud services enabled **continuous ingestion and reaction**.
- Industrial engineers recognize this immediately: alarms, process drift, predictive maintenance, energy balancing.
- Data became part of the operational fabric rather than a post hoc mirror of it.



From dashboards to intervention pipelines

Generation 3 changed the KPI of data teams

Old KPI

How much data did we ingest and how many reports did we publish?

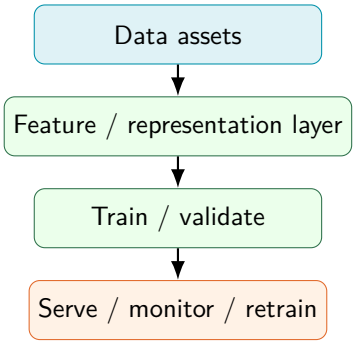
New KPI

How quickly can we detect, explain, and respond to changing reality?

- This is why observability, lineage, and low-latency architectures became strategic.
- It is also why many organizations discovered that **more data without process ownership** still does not improve outcomes.

Generation 4: Machine Learning (ML)-native data systems

- Models moved from research artifacts to production assets.
- Feature stores, experiment tracking, model monitoring, and feedback loops became part of the stack.
- Multimodal data entered the mainstream: time series, vision, text, telemetry, and simulation outputs.
- The execution problem now included **training, serving, evaluation, and drift management**.



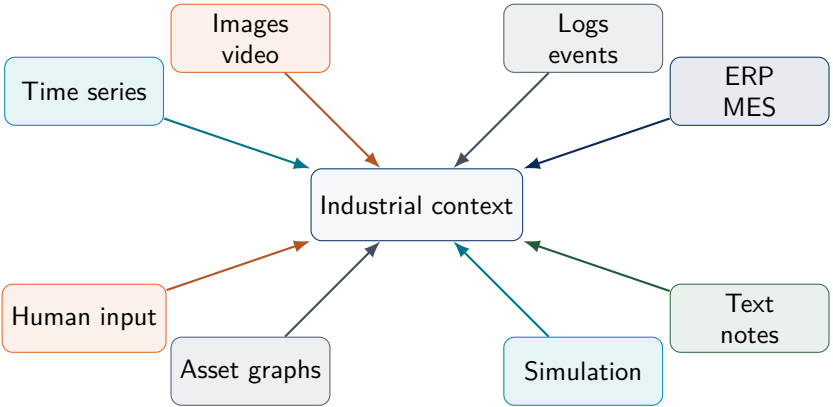
Generation 5: AI-executed systems

The newest shift is not just AI models inside data systems. It is AI participating in execution itself.

- Agents summarize streams, propose actions, generate queries, inspect anomalies, and orchestrate workflows.
- Natural language becomes a control surface, but only if grounded in reliable context.
- The winning architecture is not "ask the model anything." It is **constrained autonomy over trusted data products and workflows**.

AI without data discipline = elegant nonsense at industrial speed

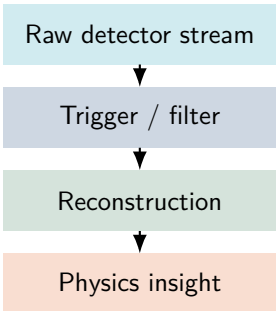
What industrial data really looks like



- Big data execution in industry is fundamentally **multimodal and socio-technical**.
- If the platform only understands tables, it sees only a fraction of reality.

CERN/CMS as an extreme data laboratory

- At CMS, instruments generate an overwhelming volume of collision information.
- We cannot keep everything. We must decide, filter, reconstruct, and validate in layered stages.
- That is why high-energy physics is a useful metaphor for industry: **selection under pressure**.



Translation

A factory also cannot “keep and analyze everything later” if the process consequence happens now.

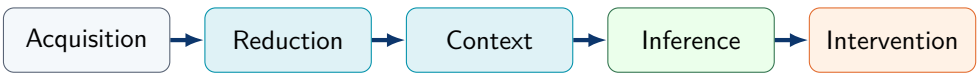
Lesson from CMS: execution is layered, not monolithic

1. **Fast path**: identify what must happen in real time.
2. **Refinement path**: enrich, calibrate, reconstruct, and contextualize.
3. **Science path**: support deeper analysis and discovery offline.

Industrial interpretation

Separate the control loop, the operational intelligence loop, and the strategic optimization loop.
Do not force one architecture to serve all three equally well.

From petabytes to decisions: the pattern worth stealing



- **Acquisition** alone is not a strategy.
- Value appears when systems intentionally reduce data to the **most decision-relevant signal**.
- In AI age, context is the difference between automation and expensive guessing.

Real-time selection matters more than ever

- Compute is growing, but data growth and complexity often grow faster.
- The industrial equivalent of a trigger is everywhere:
  defect detection, abnormal vibration, unsafe state, bad batch, supply disruption.
- We need policies for **what to keep, what to summarize, what to escalate, and what to ignore**.

Design question

If your bandwidth or attention were cut by 90%, what would you still need to preserve to act well?

Data quality monitoring is where AI earns trust

- In scientific and industrial systems alike, a bad input stream can look like a novel event.
- AI is powerful for anomaly detection, but only when paired with domain-aware monitoring and human review paths.
- Start with **quality monitoring, consistency checking, and triage** before fully autonomous control.

High-trust AI use

Detect, rank, compare, explain.

High-risk AI use

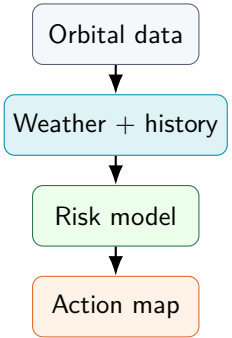
Act without guardrails on weakly grounded signals.

How science lessons transfer to Industry 4.0

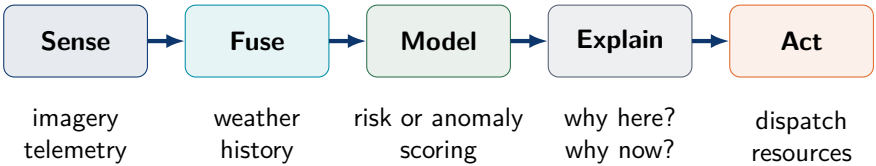
Scientific setting	Industrial setting	Shared execution principle
Trigger systems	Alarm and control thresholds	Decide early, decide safely
Calibration pipelines	Sensor alignment and maintenance	Context changes the meaning of data
Distributed analysis	Multi-site operations and suppliers	Federate computation around shared semantics
Data quality monitoring	Process quality monitoring	Trust must be continuously earned

Sharpsat: from orbit to actionable agronomy

- Satellite imagery is a perfect example of **high-volume, high-variability data that only matters when converted to local intervention**.
- Raw pixels are not value. Value is: disease risk map, stress hotspot, treatment timing, field prioritization.
- This is still Industry 4.0 logic, even if the "factory" is a vineyard.



Sharpsat pipeline as an Industry 4.0 template



Industrial AI adoption accelerates when outputs are spatially, temporally, and operationally specific.

What the startup world taught me about data execution

- No customer pays for a data lake. They pay for reduced loss, higher yield, lower downtime, safer operations.
- The hardest engineering work is often **last-mile trust**: explaining the recommendation in domain language.
- Startup pressure exposes architecture truth quickly:
  if the loop is too slow or too opaque, the product does not survive.

Business value appears at the point of intervention

The real shift: from dashboards to decision loops

Dashboard era

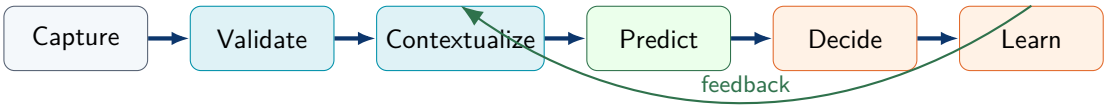
Observe
Explain after the fact
Human decides manually
Action is external to the data platform

Decision-loop era

Observe continuously
Predict and rank options
Human or machine intervenes
Feedback updates the system

Execution maturity is measured by feedback closure.

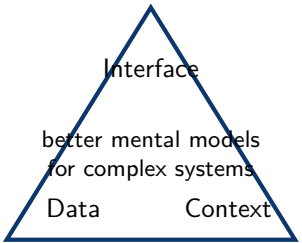
A practical industrial data life cycle



- Most organizations invest heavily in capture and very lightly in learn.
- AI age rewards the organizations that make the loop closed, measurable, and repeatable.

Why human-facing software matters in the AI data stack

- When data becomes spatial, temporal, and multimodal, the interface matters.
- Educational tools, analysis environments, dashboards, and media workflows all shape how people understand system state and possible futures.
- In other words: good software reduces the cognitive distance between **data** and **judgment**.



Humans stay in the loop, but their role changes

- The goal is not to remove people. It is to move them from **manual polling** to **judgment, exception handling, and policy setting**.
- In the AI age, the operator increasingly supervises a hierarchy of automated routines, models, and agents.
- This requires new ergonomics: explanation, override paths, simulation, and accountability.

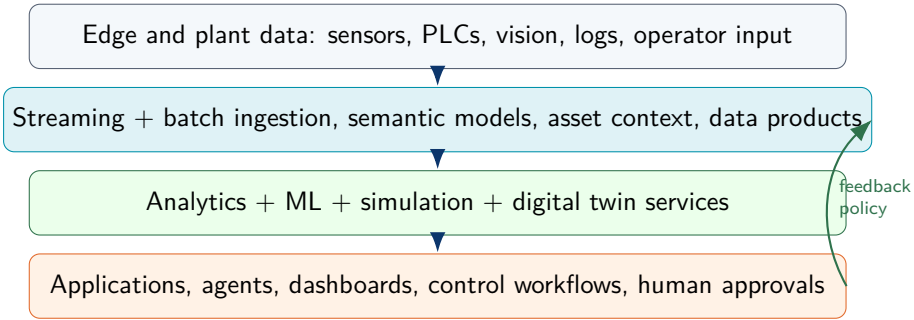
Engineers will manage confidence, not just machines

AI agents on top of industrial data: where they fit

Agent role	Good use cases	Guardrails
Analyst agent	summarize shifts, compare runs, answer grounded questions	read-only context, provenance
Triage agent	rank anomalies, route tickets, suggest checks	confidence thresholds, escalation policy
Orchestrator agent	assemble workflows, generate reports, trigger APIs	bounded tools, approval gates

Agents work best when the environment is structured, observable, and permissioned.

A blueprint for big data execution in the age of AI



- Notice the center of gravity: **semantic context and governed data products**.

The non-negotiables: trust, latency, security, cost

Trust stack

lineage, validation, explainability, monitoring, override paths

Security stack

identity, segmentation, least privilege, safe tool access

Latency stack

edge processing, event routing, caching, prioritization

Cost stack

store less blindly, summarize better, move compute intentionally

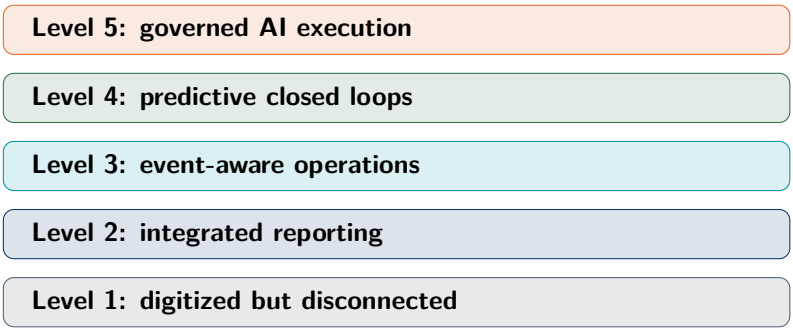
In AI age, architecture is governance expressed in software.

Seven anti-patterns I see repeatedly

1. Calling a reporting warehouse a real-time operating platform.
2. Treating AI as a UI layer on top of broken semantics.
3. Collecting everything but owning no decision loop.
4. Centralizing all intelligence and starving the edge.
5. Ignoring human override and explanation pathways.
6. Optimizing model accuracy while neglecting data quality drift.
7. Launching pilots with no plan for integration into MES, ERP, or maintenance workflows.

Most failures are architectural, not algorithmic

Adoption maturity is different from the generational shifts



- The right next step is not “jump to Level 5.” It is remove the most binding execution bottleneck.

The team changes too

Then	Now	Next
DBA	platform engineer	AI workflow designer
ETL developer	ML engineer	simulation engineer
BI analyst	domain data product owner	human-AI operations lead

The new scarce skill is translating between physics, process, data, and action.

If I were starting tomorrow in an industrial organization

1. Map the top three high-value decision loops.
2. Identify their required latency, confidence, and ownership.
3. Build a semantic layer around assets, states, and events.
4. Instrument feedback: what happened after the recommendation?
5. Add AI where it reduces cognitive load or decision latency, not where it only looks modern.

Start from decisions, not from tools

Three concrete use cases that justify the architecture

Predictive maintenance	Quality intelligence	Energy optimization
from vibration and temperature to work orders and spare planning	from vision and process data to defect prevention and parameter tuning	from load patterns to dynamic scheduling and cost-aware operation

All three need the same thing: fast, contextual, trustworthy execution.

Five takeaways

1. Big data evolved from storage systems to **decision systems**.
2. In Industry 4.0, execution quality matters more than raw data volume.
3. Science, satellite analytics, and human-facing software all point to the same truth: **context is everything**.
4. AI creates value when placed inside governed loops, not above disconnected silos.
5. The future belongs to organizations that can combine **speed, trust, and feedback**.

Closing thought

The generational evolution of big data is really a story about shrinking the distance between reality and intelligent action.

The AI age rewards organizations that can learn while they operate.