

howest

Physical AI: Teaching Machines to Act in the Real World

Jens Krijgsman | Howest AI Lab



Language was the easy part.

Ask a chatbot

"Write a poem about folding laundry."

OK

Ask a robot

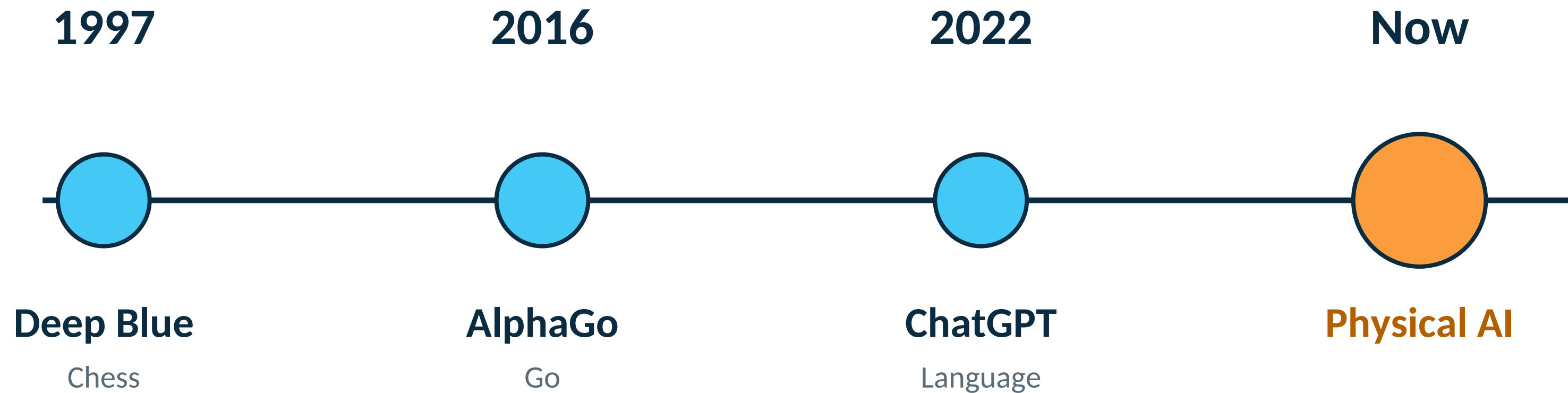
"Fold this laundry."

?



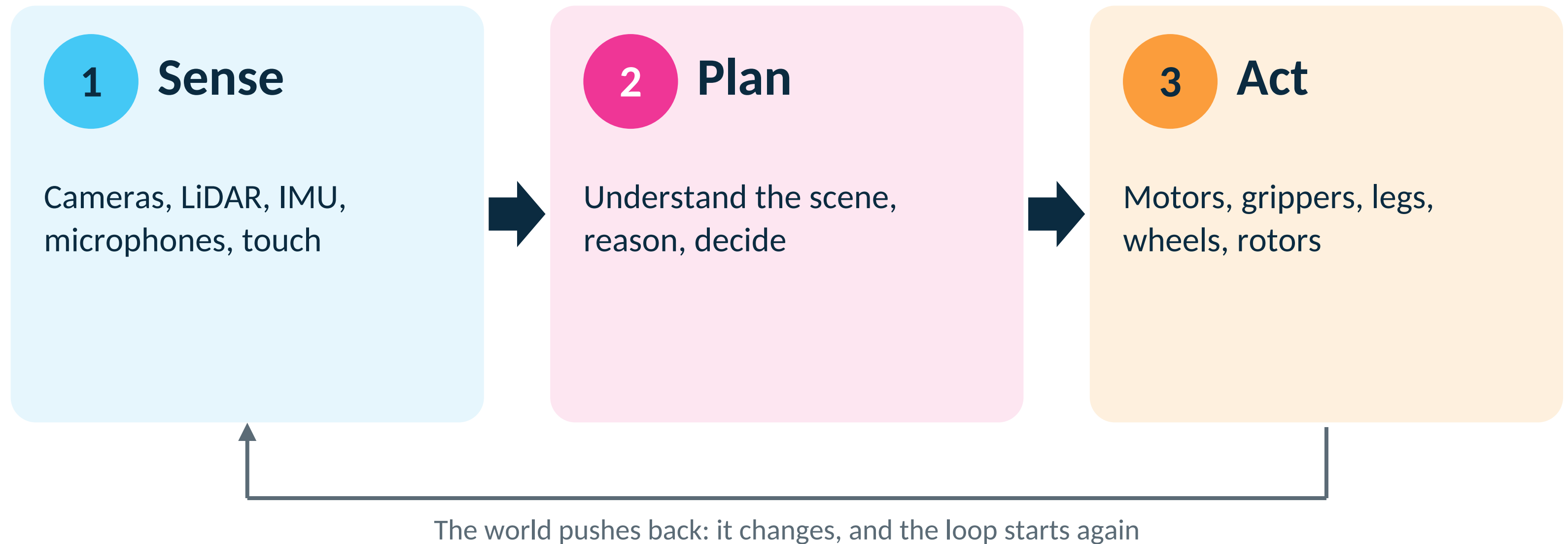


A timeline up until now





Physical AI = Sense, Plan, Act





Why now? Three things converged

01

Foundation models

Vision-language models bring common sense: objects, scenes, instructions in plain language.

02

Simulation at GPU scale

Thousands of virtual robots learn in parallel before one real motor spins.

03

Hardware you can buy

Quadrupeds and humanoids moved from lab curiosity to purchasable platforms.

Same recipe as the AI boom: data, hardware, algorithms. **For robots, the scarce ingredient is data.**

howest

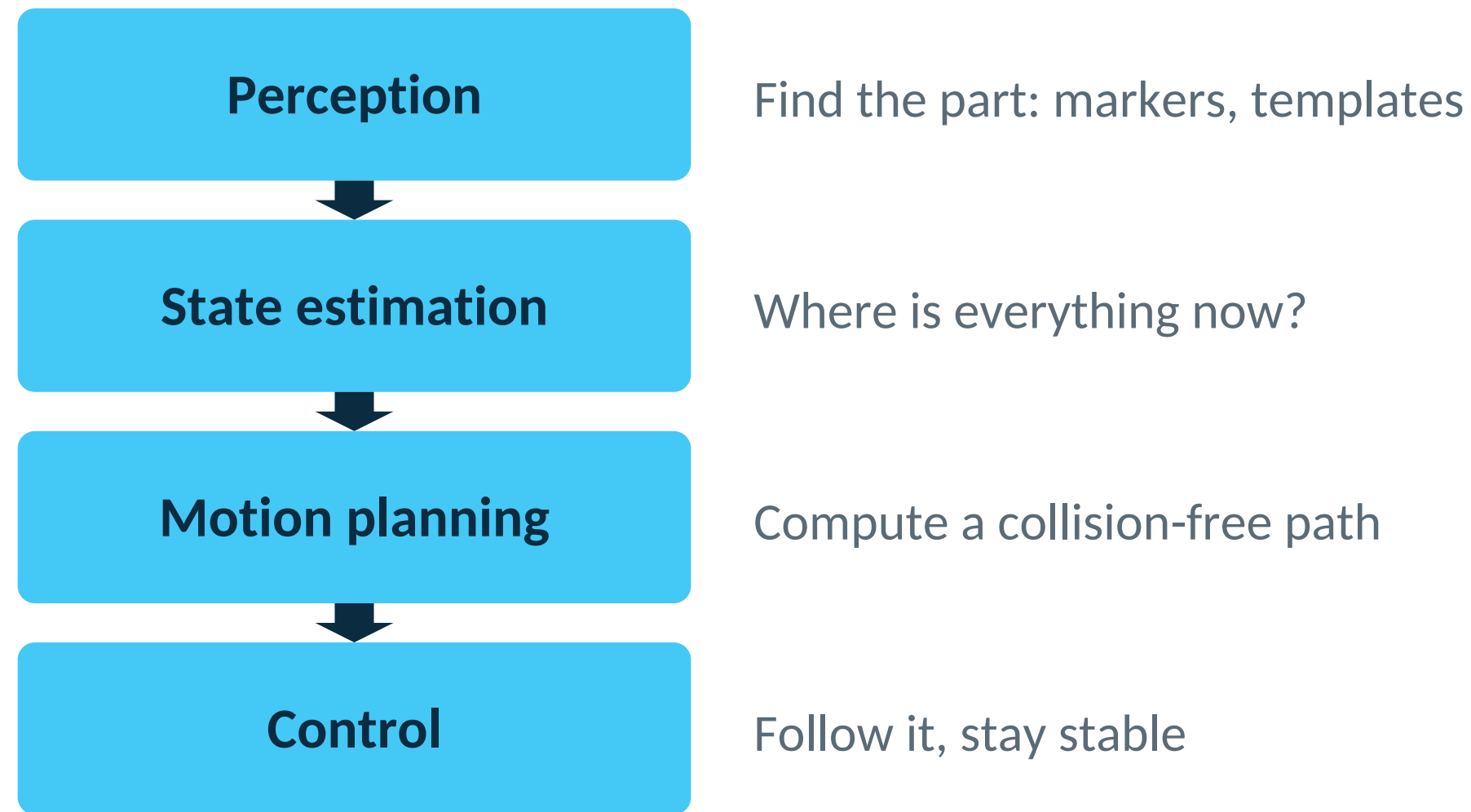
PART 1

The brain

From hand-coded pipelines to foundation models



Classic robotics: a pipeline you program



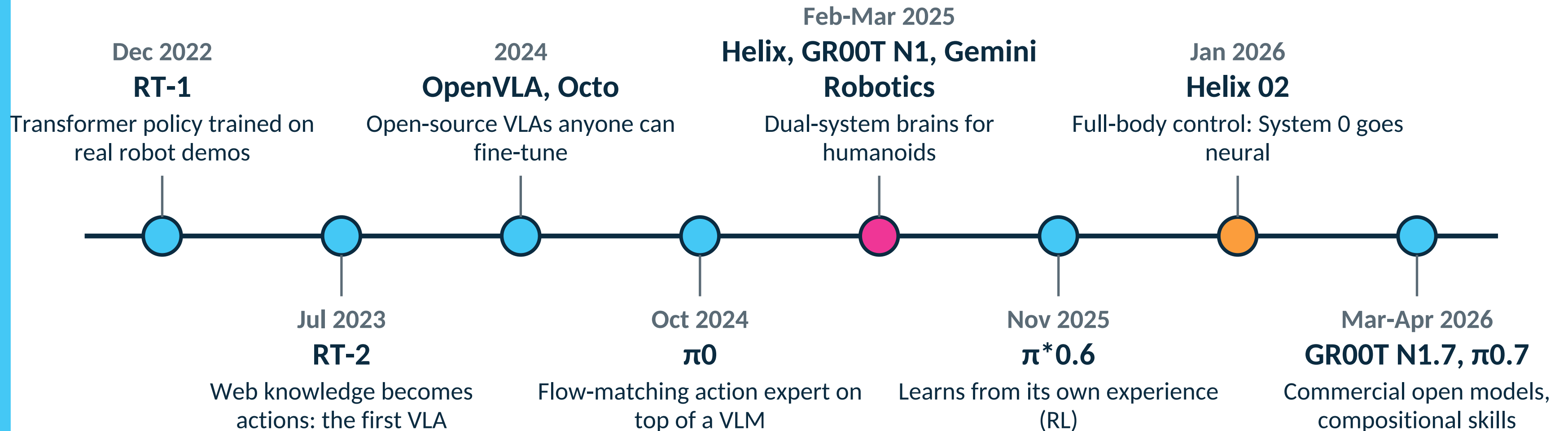
Predictable and fast. But every box is hand-built for one setup.

Fine in a cage. Brittle when the product mix changes.



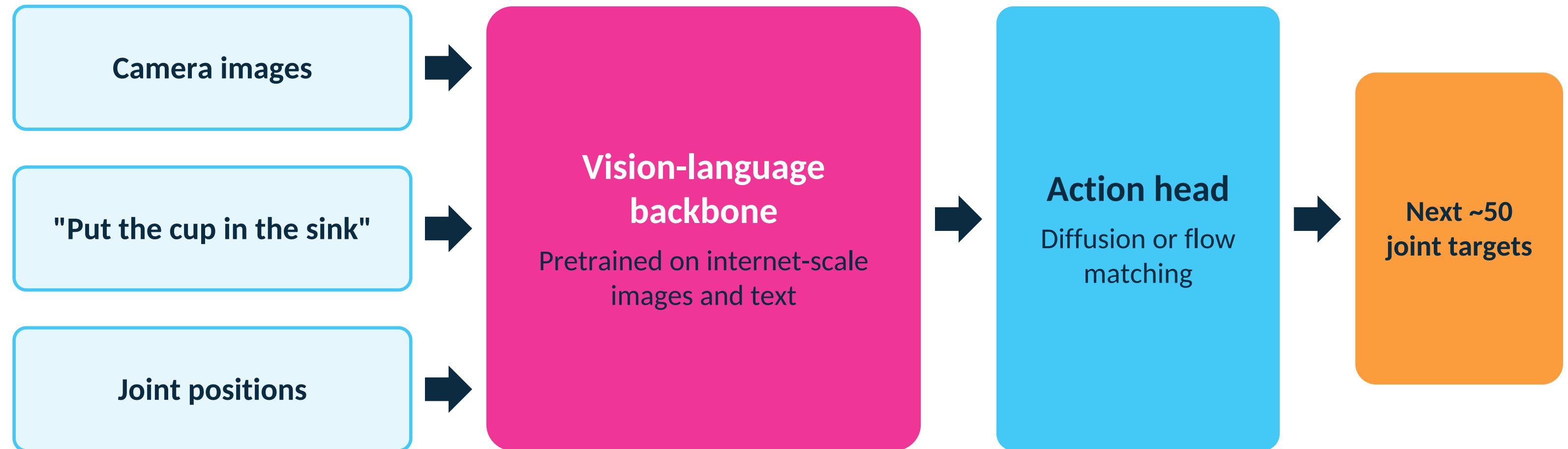


Four years: from RT-1 to $\pi 0.7$





A VLA: a language model that moves



Language models output words. VLAs output motion.

Vision-Language-Action model: the building block behind almost every modern robot brain





Why you can't just plug in ChatGPT

Speed

A large model thinks in seconds. A walking robot has to correct itself every few milliseconds.

Consequences

A wrong sentence is a typo. A wrong action breaks a gripper, a product, or hurts a person.

Grounding

Words describe the world. Motors need forces, positions and contact.

So we split the thinking by timescale

howest

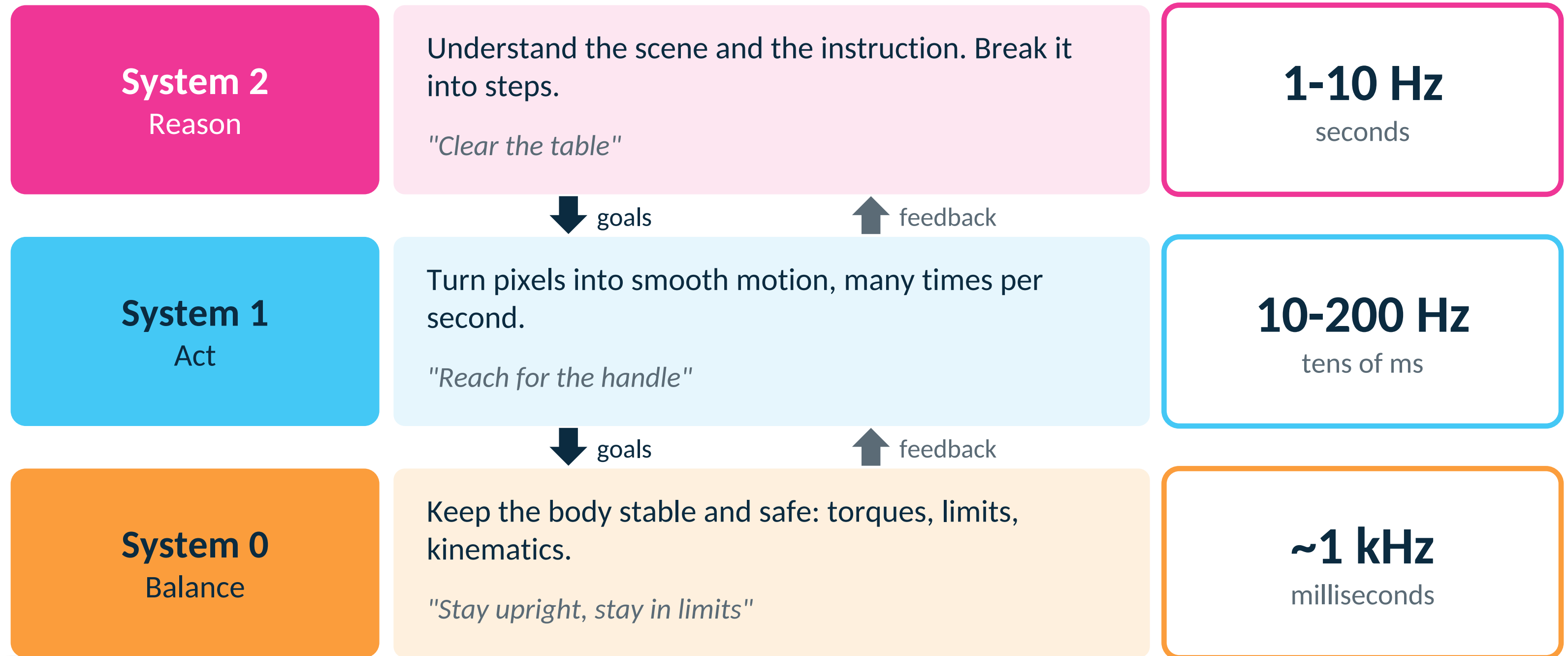
PART 2

Thinking fast and slow

System 2, System 1 and System 0 for robots



Three layers, three clocks





System 0: the body

1 kHz

control loop: balance and torque,
every millisecond

10 M

parameters: tiny next to a 7B+
language model

109,504

lines of hand-written C++ replaced
by one neural prior

Figure Helix 02 (Jan 2026): trained on 1,000+ hours of human motion, in 200,000+ parallel simulations

Classical System 0

PID, impedance control, model-predictive control. Proven, certifiable, and what sits under most learned policies today.

Learned System 0

A neural prior of how people move while staying balanced, trained in simulation with domain randomisation.



System 1: the reflexes

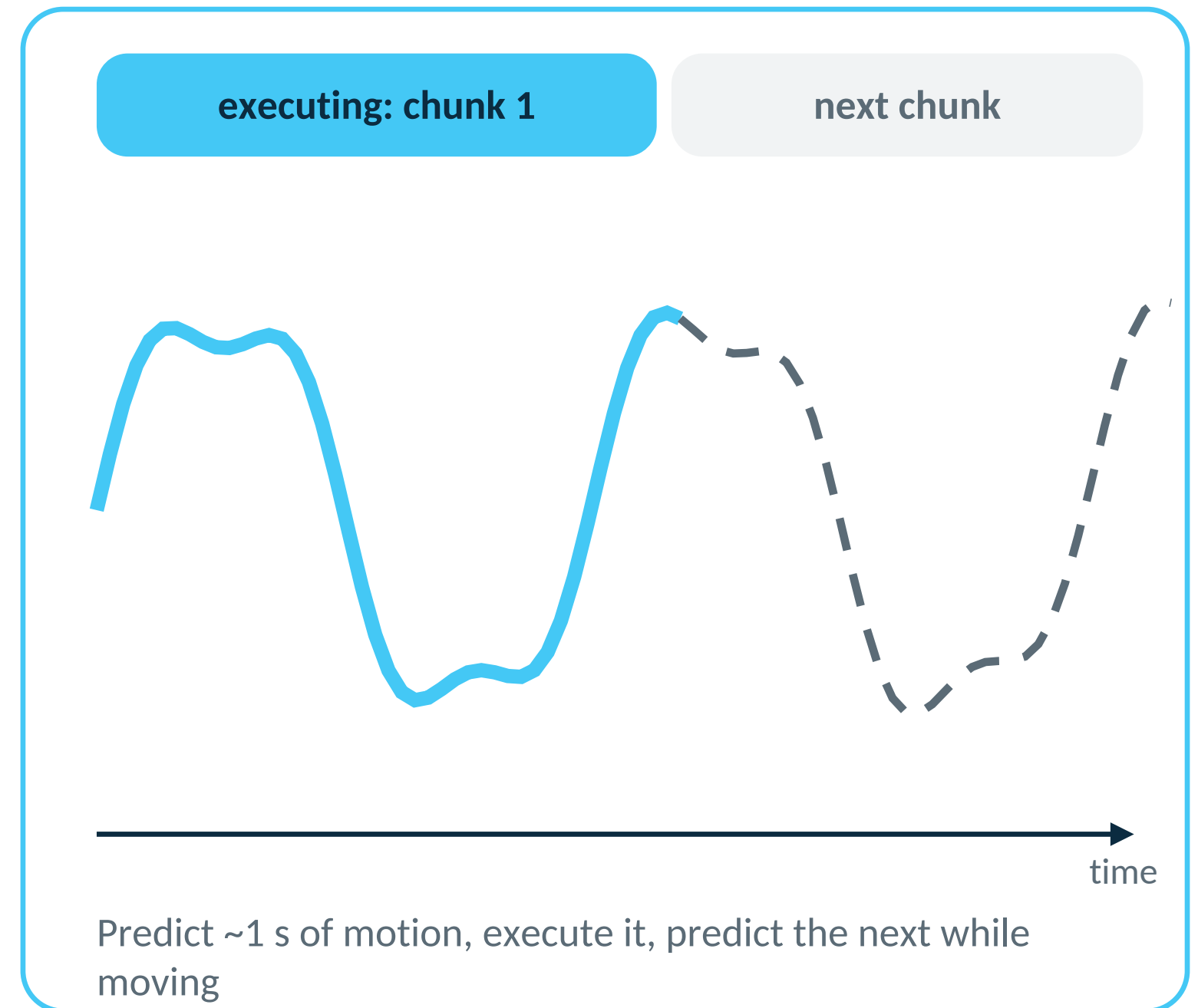
- 1 Pixels (and touch) in, joint targets out
- 2 Diffusion or flow matching generates a smooth chunk of actions
- 3 Runs 10-200 times per second

In the wild

π 0.5: VLM + flow-matching action expert

GR00T N1: diffusion transformer

Helix S1: 200 Hz, palm cameras and touch





System 2: the planner

1

Reads the scene and the instruction

A vision-language model, trained on web-scale data

2

Outputs goals, not joint angles

Sub-goals in language or a latent code

3

Slow and expensive

A few Hz is fine: plans change slower than physics

In the wild

Gemini Robotics-ER 1.5, the VLM inside GR00T, Figure Helix S2

"Clear the table and load the dishwasher"



1 Pick up the plate

2 Walk to the dishwasher

3 Open the door

4 Place the plate

5 Repeat for the cups



System 0 on the factory floor

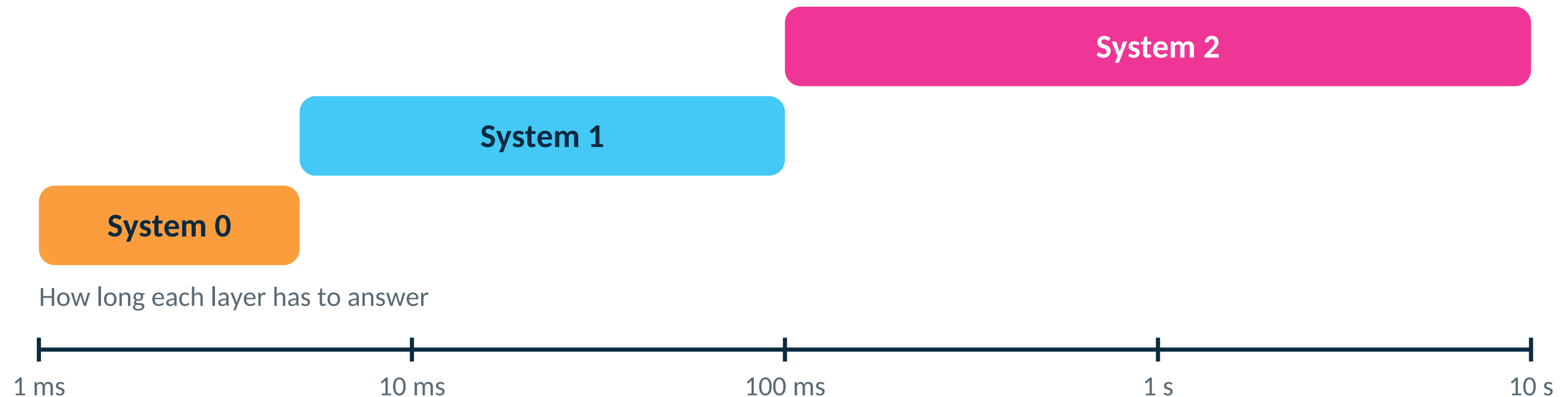
	System 2	System 1	System 0
Humanoid Helix 02	VLM sets semantic goals	Visuomotor policy, 200 Hz	Learned whole-body controller, 1 kHz
Industrial robot arm ABB, UR, Fanuc	Optional VLM planner	Vision-guided policy sends waypoints	Vendor controller: servos, safety stop
Quadruped Unitree GO2	VLM scene understanding	Inspection / navigation policy via ROS 2	Built-in locomotion controller
Wheeled robot AgileX Limo	VLM scene understanding	Navigation stack or policy	Motor drivers and base controller

In industry System 0 is usually already there: certified, boring, reliable. We build Systems 1 and 2 on top.



Why the layers stay

Slow thinking cannot hold a balance. Fast reflexes cannot plan a dishwasher.



Bandwidth

Thinking cannot keep up with physics

Safety

Limits and e-stops live in deterministic code

Sim-to-real

Low-level loops absorb model errors



One model or three layers?

	System 2 (reason)	System 1 (act)	System 0 (body)	Access
Figure Helix 02	VLM sets semantic goals	Visuomotor policy, 200 Hz	Learned, 1 kHz	Closed
NVIDIA GR00T N1.x	Vision-language module	Diffusion transformer	Robot's own controller	Open weights
Gemini Robotics 1.5	ER model orchestrates, calls tools	VLA that thinks before it acts	Platform controller	ER: API. VLA: partners
π 0.5 / π 0.7	Same model predicts sub-tasks	Flow-matching action expert	Robot's own controller	openpi (π 0, π 0.5)

Different packaging, same insight: separate what to do, how to move, and how to stay stable.



Beyond three layers: what comes next

1

Learning on the job

$\pi^*0.6$ improves with real-world experience: demonstrations, human corrections, then autonomous RL.

2

Composing skills

$\pi0.7$ recombines known skills for tasks it never saw, even on a robot it was not trained for.

3

World models in the loop

Predict the future in video first, then act. Research on world-action models is moving fast.

The trend: robots that keep learning after deployment.



howest

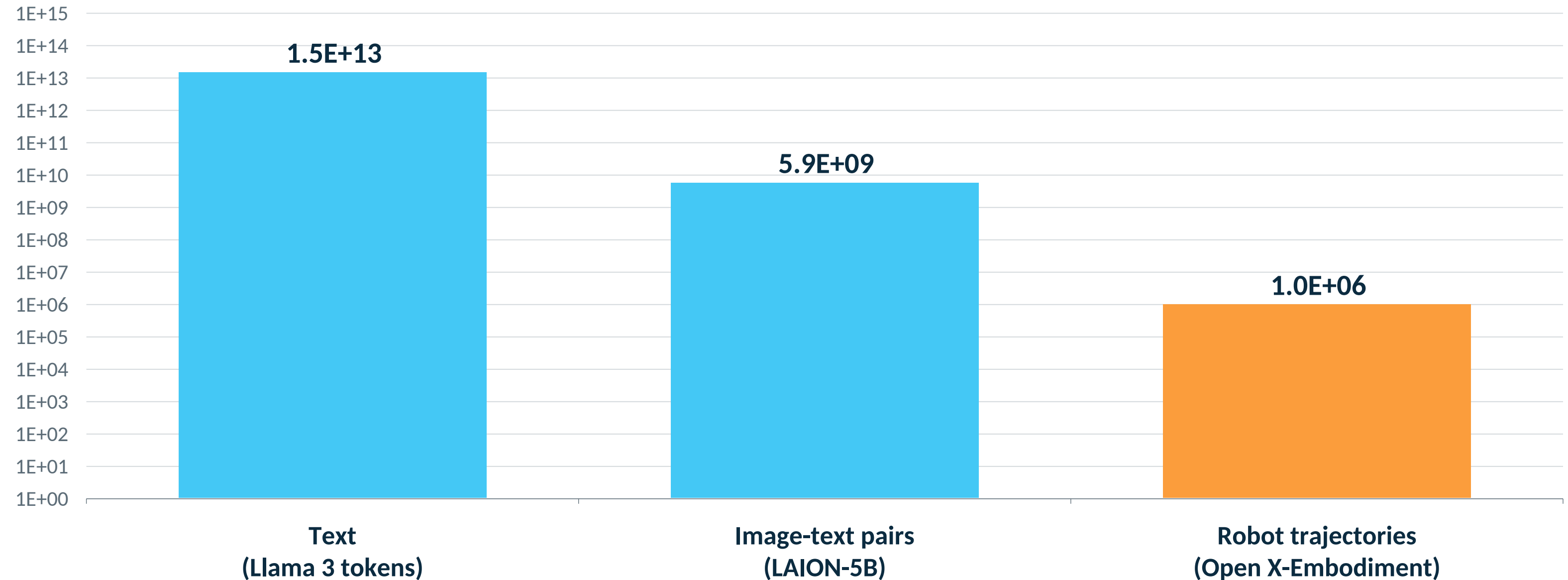
PART 3

The data problem

LLMs had the internet. Robots have to make their own data.



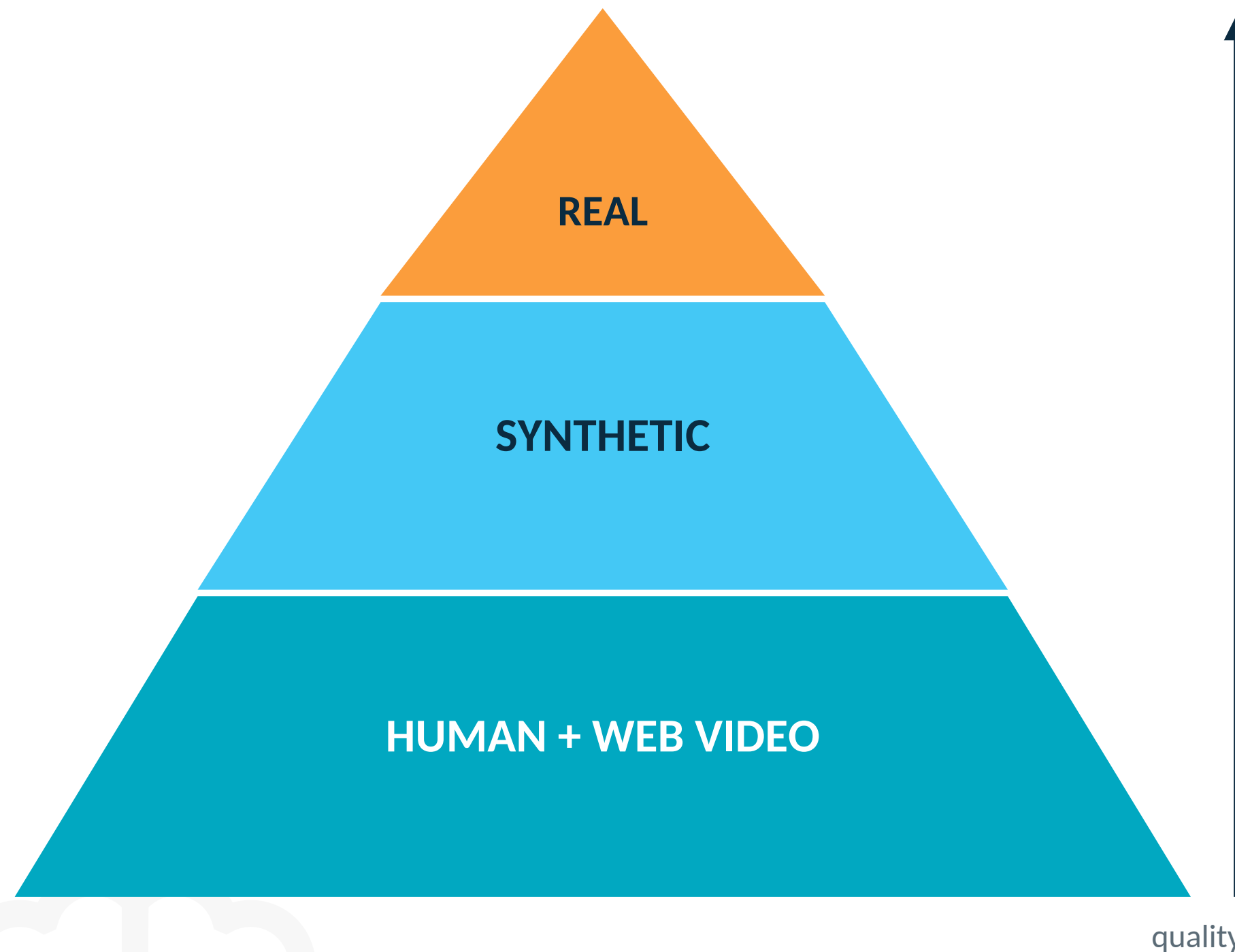
The data gap: orders of magnitude



Different units, same message: robots start 4 to 7 orders of magnitude behind.

Log scale. Sources: Meta (Llama 3, 15T tokens), LAION-5B (5.85B pairs), Open X-Embodiment (1M+ trajectories, 22 robot types).

The data pyramid



Real robot demonstrations

Scarce and expensive, but exactly on target. Teleoperation is the new annotation.

Simulation and synthetic data

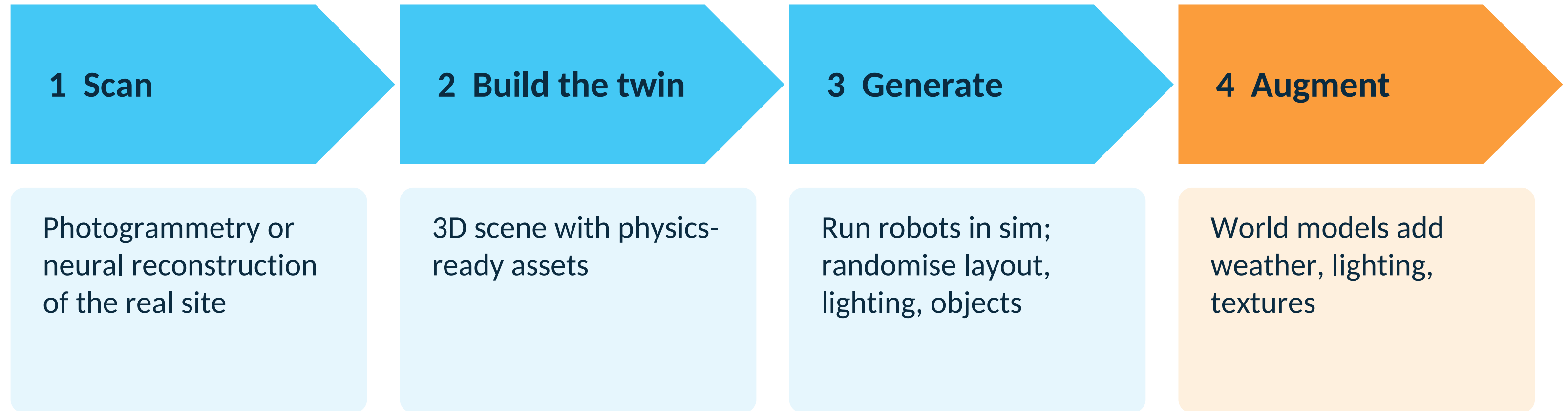
Scales with compute. Imperfect physics, perfect labels.

Human and web video

Huge, but no robot actions. Egocentric video, retargeted human motion.



Synthetic data: build a world, then multiply it



One scanned site becomes thousands of training scenes.

Tools: Isaac Sim, Omniverse NuRec, Cosmos Transfer (NVIDIA); Unity and Unreal as alternatives for the 3D twin.



howest

PART 4

Train in a thousand worlds

Simulation, digital twins and 3D training

Why simulate?



Real world

One robot

Real time

Resets by hand

Crashes cost money



Simulation

Thousands in parallel

Faster than real time

Resets in microseconds

Crashes are free

200,000+

parallel environments used to
train Helix System 0

< 20 min

for a quadruped (ANYmal) to
learn to walk on a single GPU



The simulator zoo

Physics engines

MuJoCo, Isaac Sim / Lab, Genesis, Newton 1.0

GOOD AT

Fast contact-rich RL, millions of steps per second on GPU

WEAK AT

Struggle with deformables, liquids and transparent objects

Game engines

Unity, Unreal

GOOD AT

Photoreal scenes, rich tooling, XR and human-in-the-loop testing

WEAK AT

Physics fidelity needs extra work. A strength of our HITLab

Learned world models

Cosmos, Genie 3, 1X World Model

GOOD AT

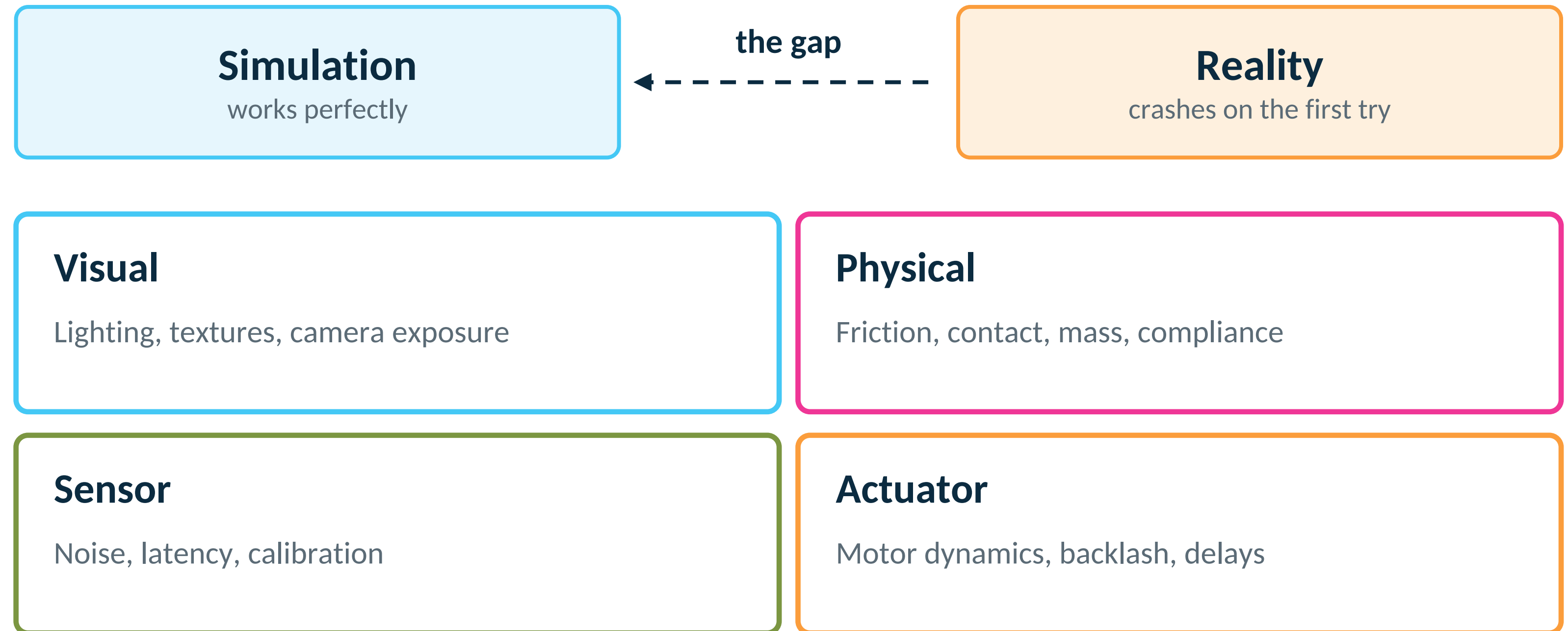
No asset creation, messy real-world visuals, policy evaluation

WEAK AT

Drift and hallucinated physics; coherent for minutes, not hours



The sim-to-real gap



Which factor dominates? That is a measurable question, and a research question in our project.



Closing the gap

1

Domain randomisation

Vary everything in sim: light, friction, mass, textures. Reality becomes just one more variation.

2

System identification

Measure the real robot and tune the simulator to match its dynamics.

3

Real-to-sim

Scan the real site, so the twin looks and measures like the place it represents.

4

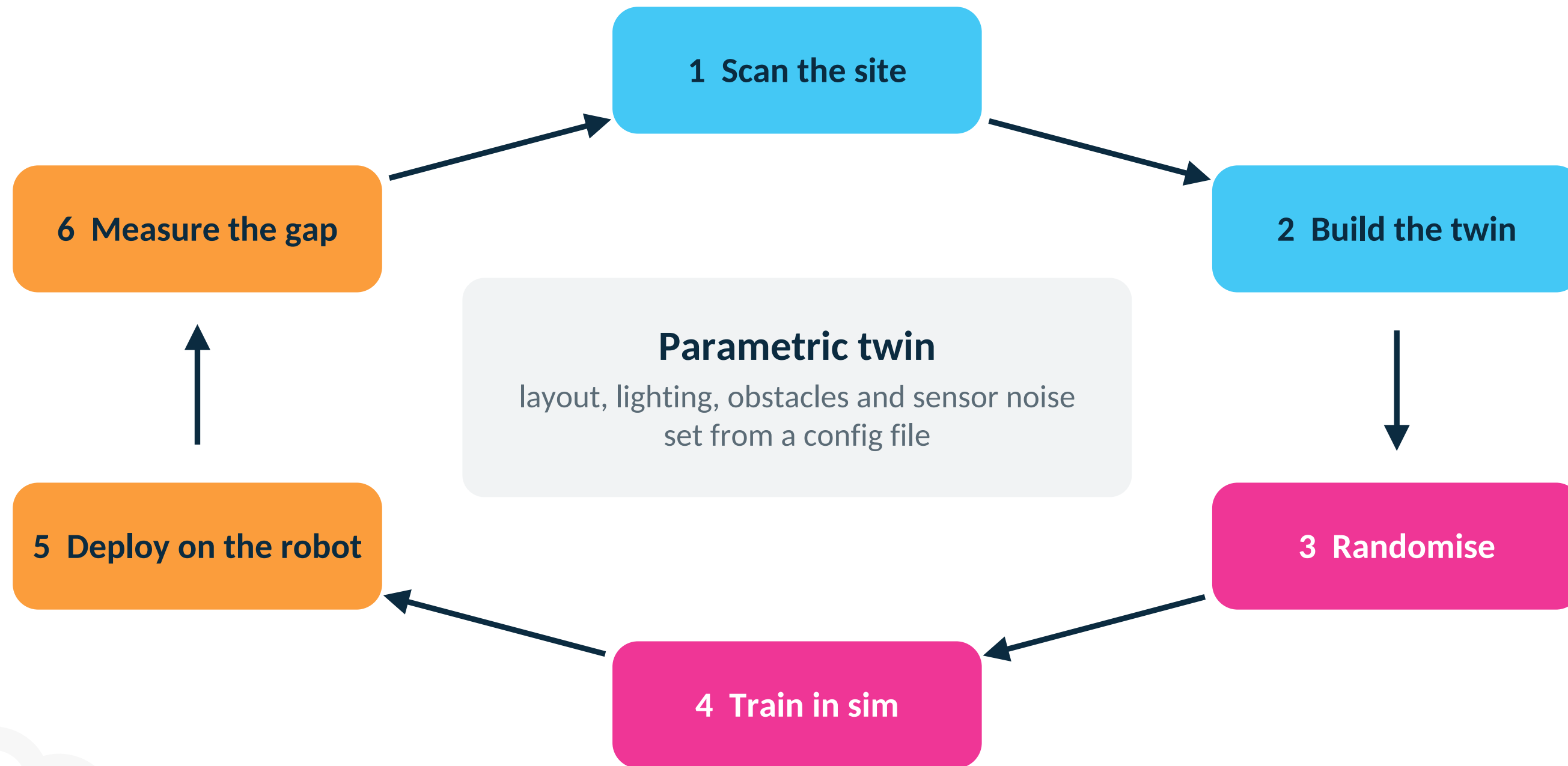
Real fine-tuning

A little real data on top of a lot of simulation, plus low-level controllers that absorb the rest.

Do not perfect the simulator. Make the policy robust.



The digital-twin training loop



howest

PART 5

It is also a software problem

The robot is the hardware. The product is the stack.



The robot software stack

Intent

Tasks, fleet orchestration, human interface

Models

System 2 VLM, System 1 VLA, perception

Middleware

ROS 2, DDS messaging, transforms, navigation

Real-time control

System 0, motor drivers, safety PLC

Hardware

Sensors, actuators, onboard compute

Our WP1 builds the Models and Middleware layers on ROS 2

Cutting across

Data and MLOps

Evaluation

Observability

Security

Safety



Where does the brain live?

On the robot

ms

Edge computer (Jetson class, even CPU) running small models such as SmolVLA (450M parameters).

Lowest latency. Limited model size.

On-prem server

10s of ms

A GPU server in the building (we have an H200) reached over Wi-Fi or 5G.

Big models, data stays local. The network is now part of the control loop.

Cloud

100s of ms

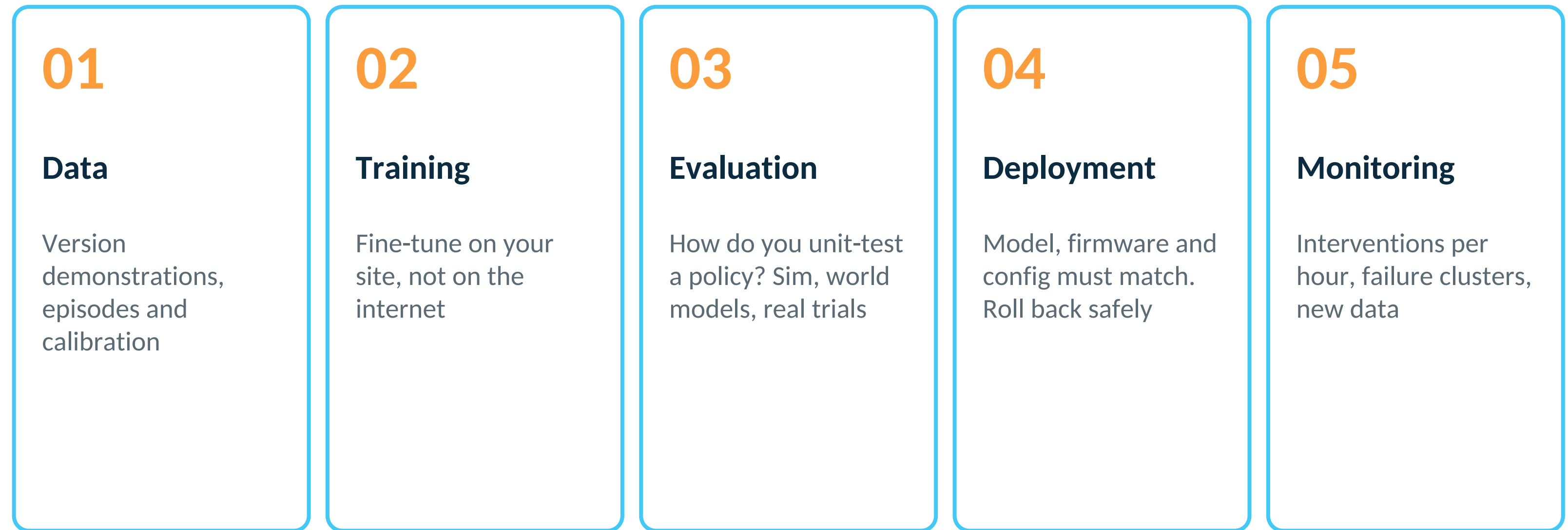
Hosted frontier models, e.g. a reasoning model behind an API.

Largest models, easy scaling. Latency, cost and data leave the site.

Trick: act while thinking. Asynchronous inference lets the robot request the next action chunk while still executing the current one (SmolVLA: about 30% faster, 2x task throughput).



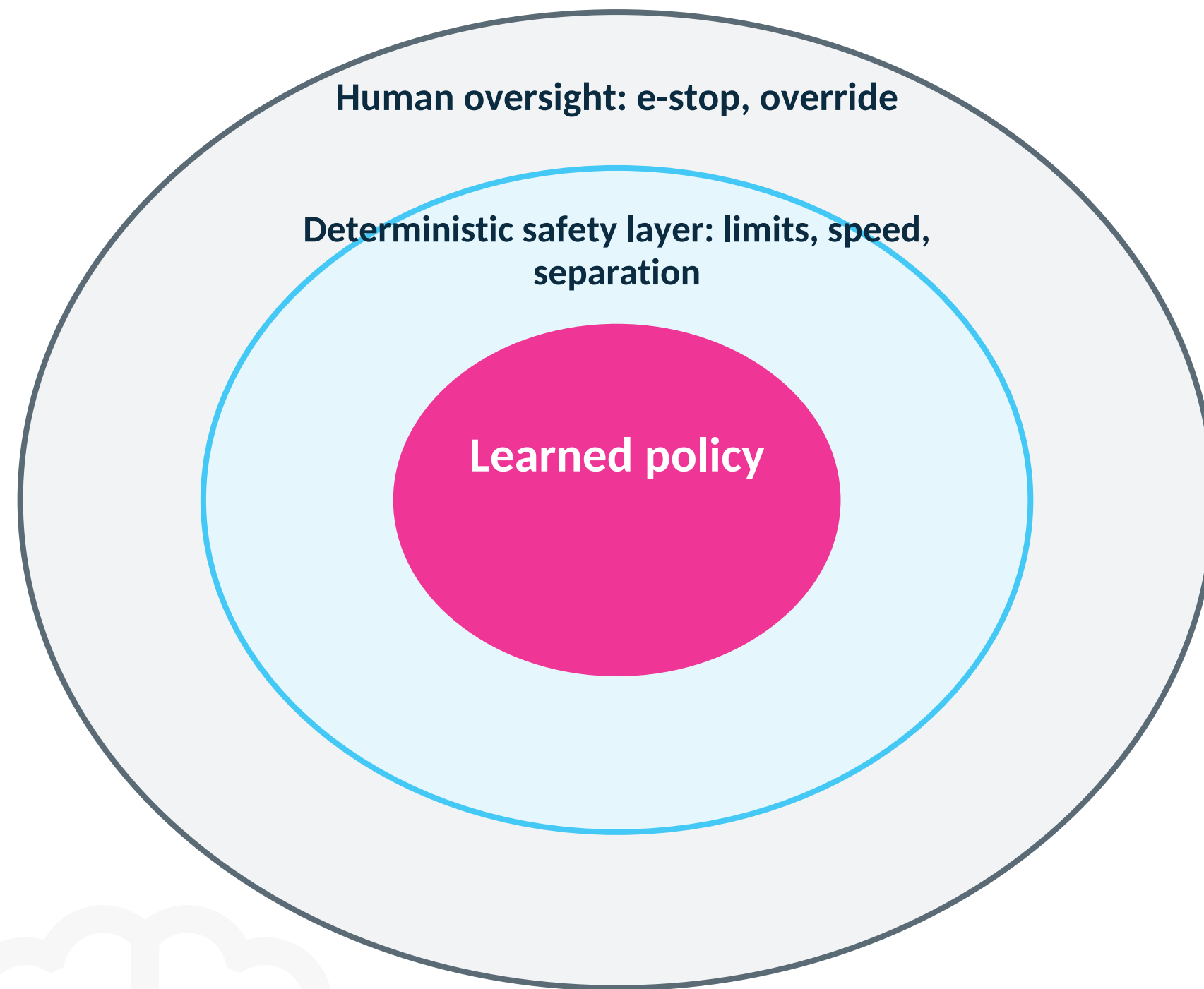
Robot MLOps: five new problems



Every failure becomes training data: the data flywheel.



Safe by architecture



The AI proposes. A deterministic layer disposes.

EU Machinery Regulation

(EU) 2023/1230 applies from 20 January 2027

Safety functions based on self-evolving machine learning are high-risk: third-party conformity assessment

Plus EU AI Act classification per system

howest

PART 6

Change the robot, or the world?

Environments, bodies and a reality check



Who adapts: the robot or the world?

HIGH VOLUME

The environment adapts to the robot

- Cages and fixtures
- Parts presented in fixed positions
- Same product for years

HIGH-MIX, LOW-VOLUME

The robot adapts to the environment

- Perception and planning needed
- Frequent product changes
- Flexibility is the product

THE PRAGMATIC MIDDLE

Robot-friendly environments

- Standard load carriers and markers
- 5G / Wi-Fi coverage, machine APIs
- A digital twin or BIM of the site



Pick the body for the job



Humanoid

Built for spaces made for human bodies: stairs, doors, tools



Quadruped

Hard-to-reach, hazardous and unstructured places: inspection



Wheeled arm

Flat floors, payload, lowest cost: most logistics

Does a robot really need legs? Only if the task does.





Reality check, 2026

56 DOF, 4 h

Atlas production version (CES 2026): 110 lb payload, battery swaps itself

30,000 / year

Hyundai target for humanoid production capacity by 2028

4 min, 61 actions

Figure Helix 02 loading a dishwasher fully autonomously



Demos show progress. Reliability, battery life, dexterity, cost and safety decide deployment.

howest

PART 7

What we do at Howest

The Physical AI project, and how you can plug in



Three labs, one robot brain

AI Lab

The brain

LLM and VLM models, agents,
MLOps, our own H200 server

Plan

HITLab

The worlds

Digital twins in Unity, AR/VR,
Sim2Real transfer

Simulate

Energy Lab

The senses

Sensors, IoT, 5G connectivity,
embedded and edge energy

Sense

Together: the PWO programme Physical AI for industry (2026-2029)





Physical AI project: three years

Year 1 Build knowledge

VLA and VLM models on our hardware

ROS 2 Sense-Plan-Act pipeline on 2 robots

Digital twin v0.1 from 3D scans

Course design for a Physical AI elective

Year 2 Go deeper

Measure and shrink the Sim2Real gap

Transfer learning between sites

Agentic planning on the quadruped

SPARK (previously Tetra) application (Feb 2028/29)

Year 3 Integrate

Multi-robot coordination

Industrial pilots with partners

Open-source release

First industrial use-case validation

We are here: month 2

Our robot lab



And waiting on a project to have more budget for new robots

Unitree GO2

Quadruped for inspection and navigation

AgileX Limo

Wheeled platform to start simple

Bitcraze Crazyflie

Small drones for the overview shot

H200 AI server

Training and inference of VLA models

3D scanner

Real sites into digital twins

5G lab

Low-latency link between robot and server



Five questions we are answering

01

Robustness

What breaks a Sense-Plan-Act system in a variable factory?

02

Sim2Real

Which simulation choices shrink the gap most?

03

Transfer

How little retraining does a new site need?

04

Agentic

Can a VLM plan and coordinate a team of robots?

05

Adoption

What decides success in real companies?

Everything from this talk, turned into experiments.



Use cases from the field



Site surveillance

A robot platform for security patrols
with Mentor Worldwide Security Services



Solar-panel inspection

Robot dog and drone working together
with GPC



Hazardous infrastructure

Inspection where people should not go
inspired by Artes Group (construction)

We start from real needs, then generalise into methods that transfer.





How we can help you

Try it

Robotics Living Lab

An open test environment where companies can try robots and models on their own use case.

Learn it

Workshops and training

Introductions first, hands-on sessions later. Plus a microdegree for professionals.

Build it

Open-source toolkit

A ROS 2 reference architecture, configs and guides to start from.

Staff it

Students and projects

Bachelor theses, internships and applied projects on your use case.

Bring us a use case. We will tell you honestly if Physical AI fits.



Take-aways

1

Three clocks

Physical AI thinks at three timescales: reason (System 2), act (System 1), balance (System 0).

2

Data is the bottleneck

Simulation, digital twins and synthetic data close the gap, if we manage the sim-to-real gap.

3

Software and environment

The stack, the safety architecture and the environment matter as much as the model.

4

A family of robots

Not one robot that does everything, but embodied systems that each solve a specific real-world problem.



howest

Let's get in touch!

Jens Krijgsman | jens.krijgsman@howest.be