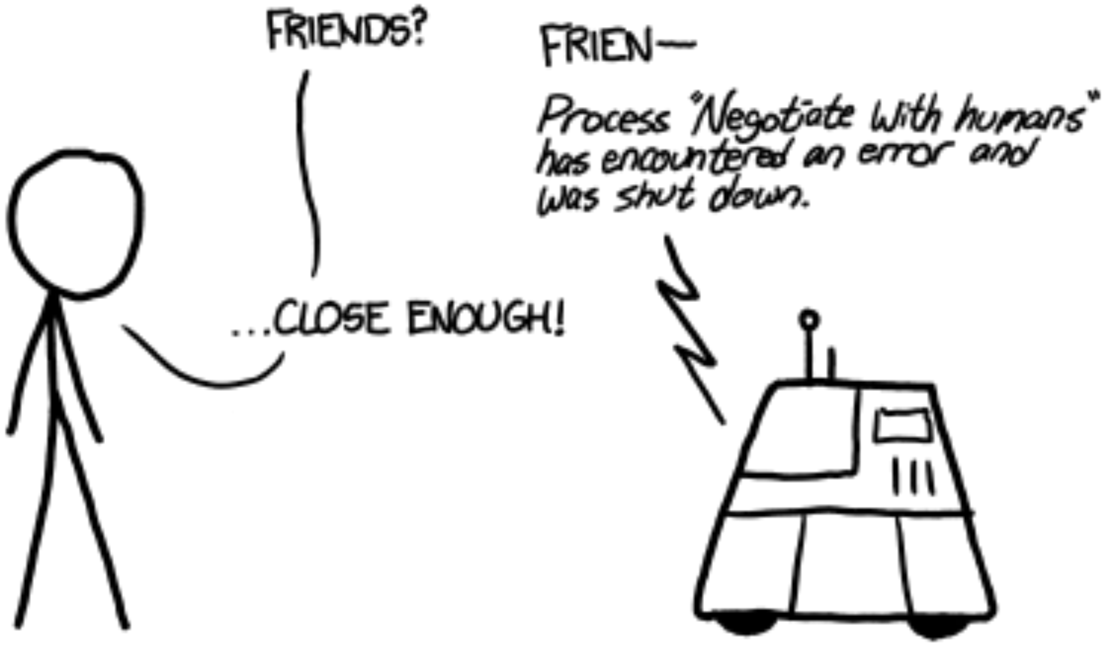
(Slides content adapted from Koen Hindriks)



Lecture 8
"What are you thinking?"
Cognitive Skills for a Robot

Kim Baraka
Assistant Professor
Social AI group

Go to **www.menti.com** and use the code **4311 2495**

Cognitive skills in humans?

processing information
decisions
predictive error
imagining
language
action planning
processing input
reasoning
memory
theory of mind
reasonable thinking

8

Go to **www.menti.com** and use the code **4311 2495**

Non-cognitive skills in humans?

emotion

digestion

reflexes

nervous system

external inputs

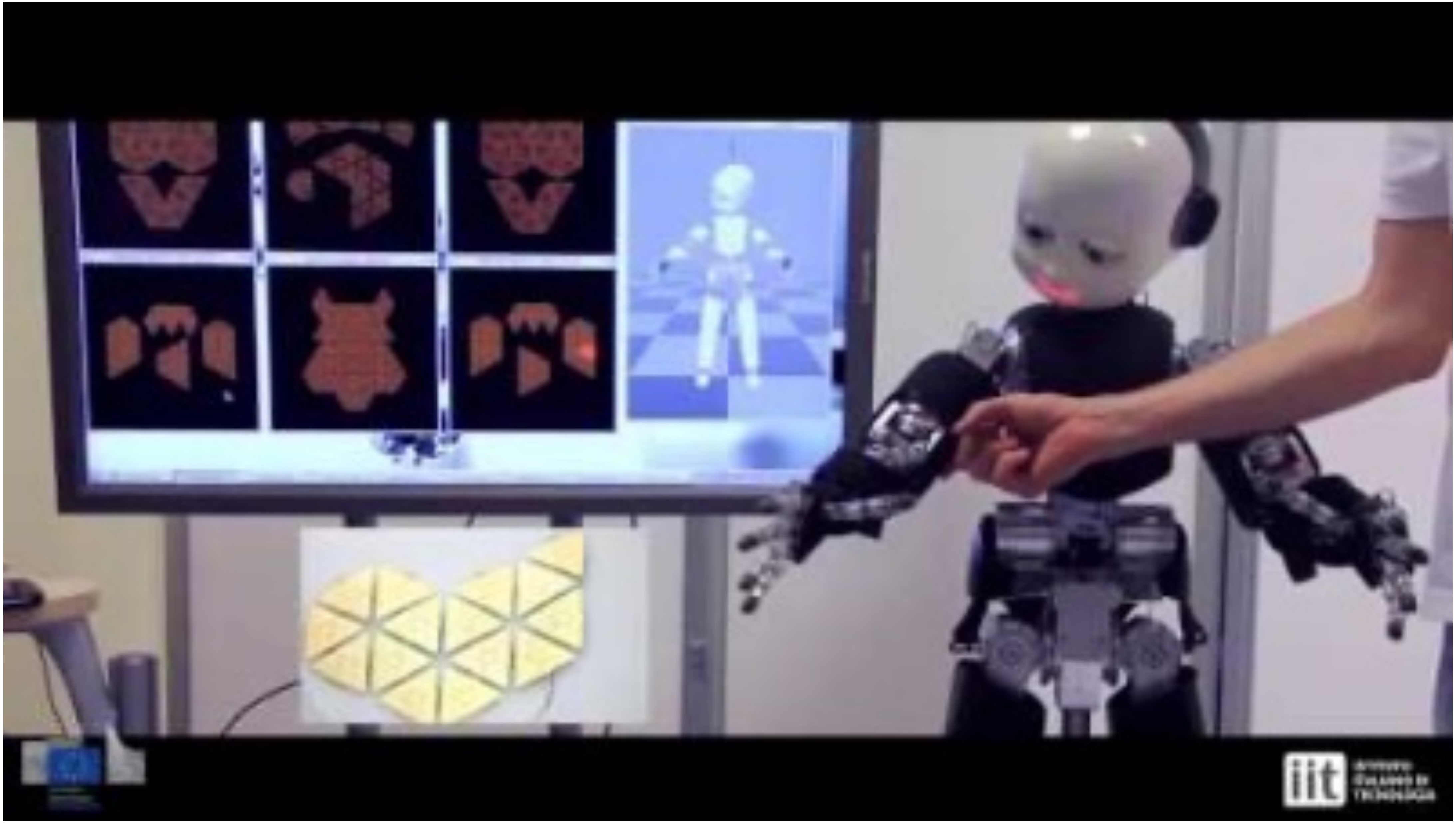
walking

6

“Cognition” is a loaded term with varying definitions

- Traditionally, “social skills” are generally considered non-cognitive
- In fact, social skills heavily rely on cognitive processes
- Social cognition: “Mental processes involved in perceiving, attending to, remembering, thinking about, and making sense of the people in our social world.” (G. B. Moskowitz)

In this lecture, we take a broad view on what counts as a cognitive skill in a robot: basically any entity or process that allows a robot to process information and communicate with other similar entities, the environment, or other agents



https://www.youtube.com/watch?v=ErgfgF0uwUo

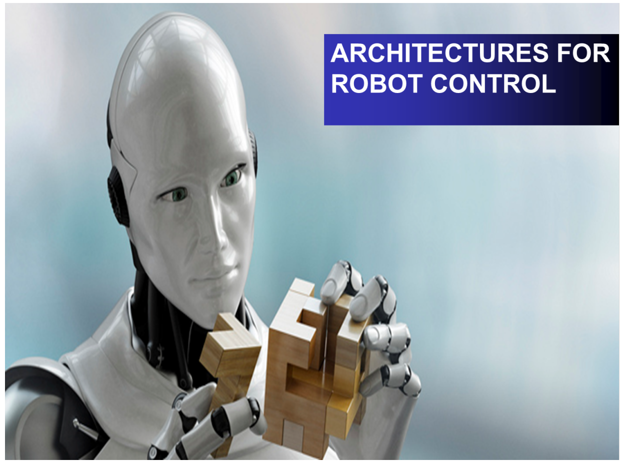
Goal: Introduce architectural issues related to integrating cognitive robot skills.

Learning Objectives:

- Able to explain key challenges of integrating robot cognitive skills
- Able to explain various robot architecture proposals
- Able to explain issues related to persistence & cloud robotics

Outline


CHALLENGES FOR
ROBOT CONTROL


ARCHITECTURES FOR
ROBOT CONTROL


PERSISTENCE &
CLOUD ROBOTICS

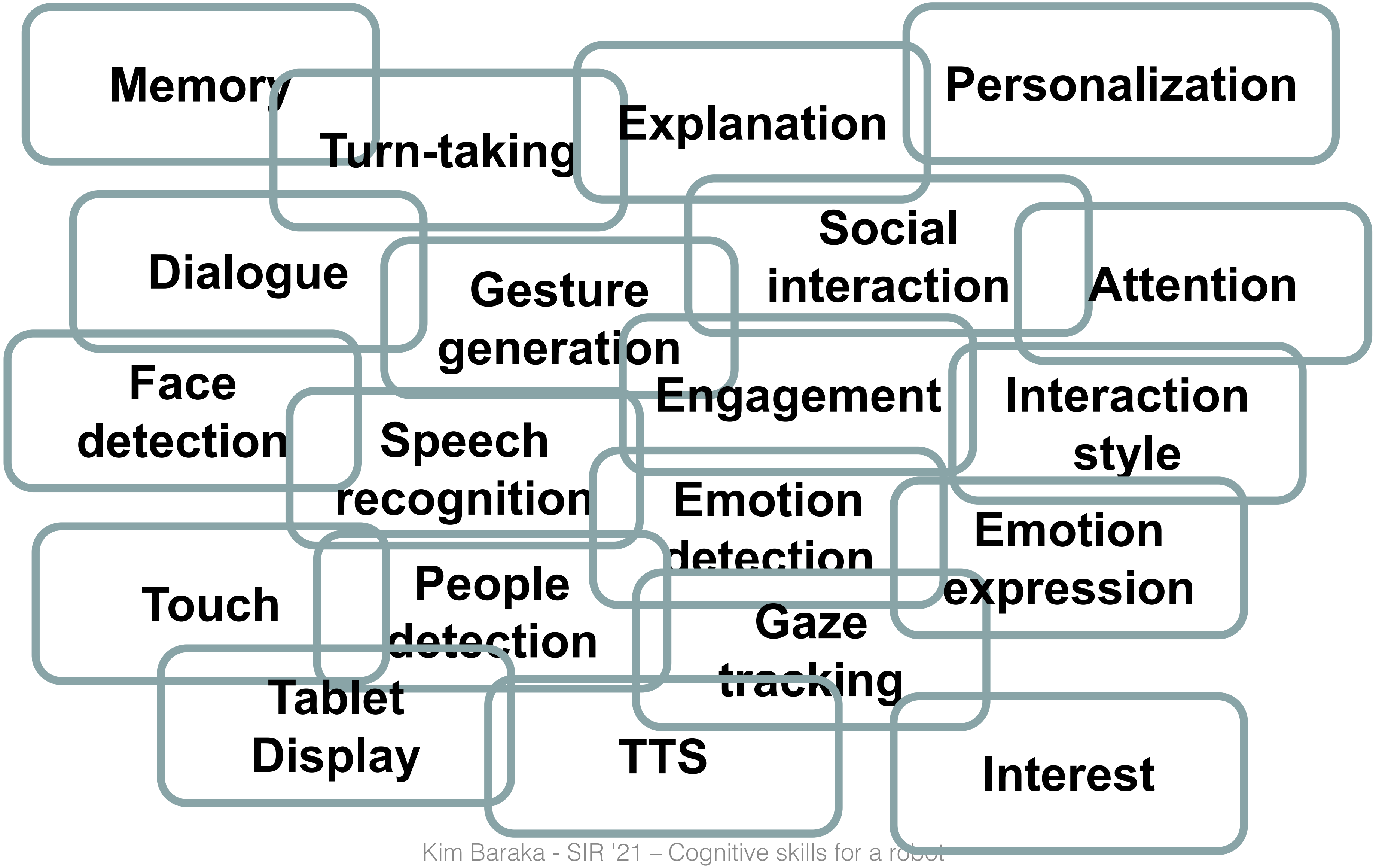
CHALLENGES FOR ROBOT CONTROL

Motivating example: social greeting

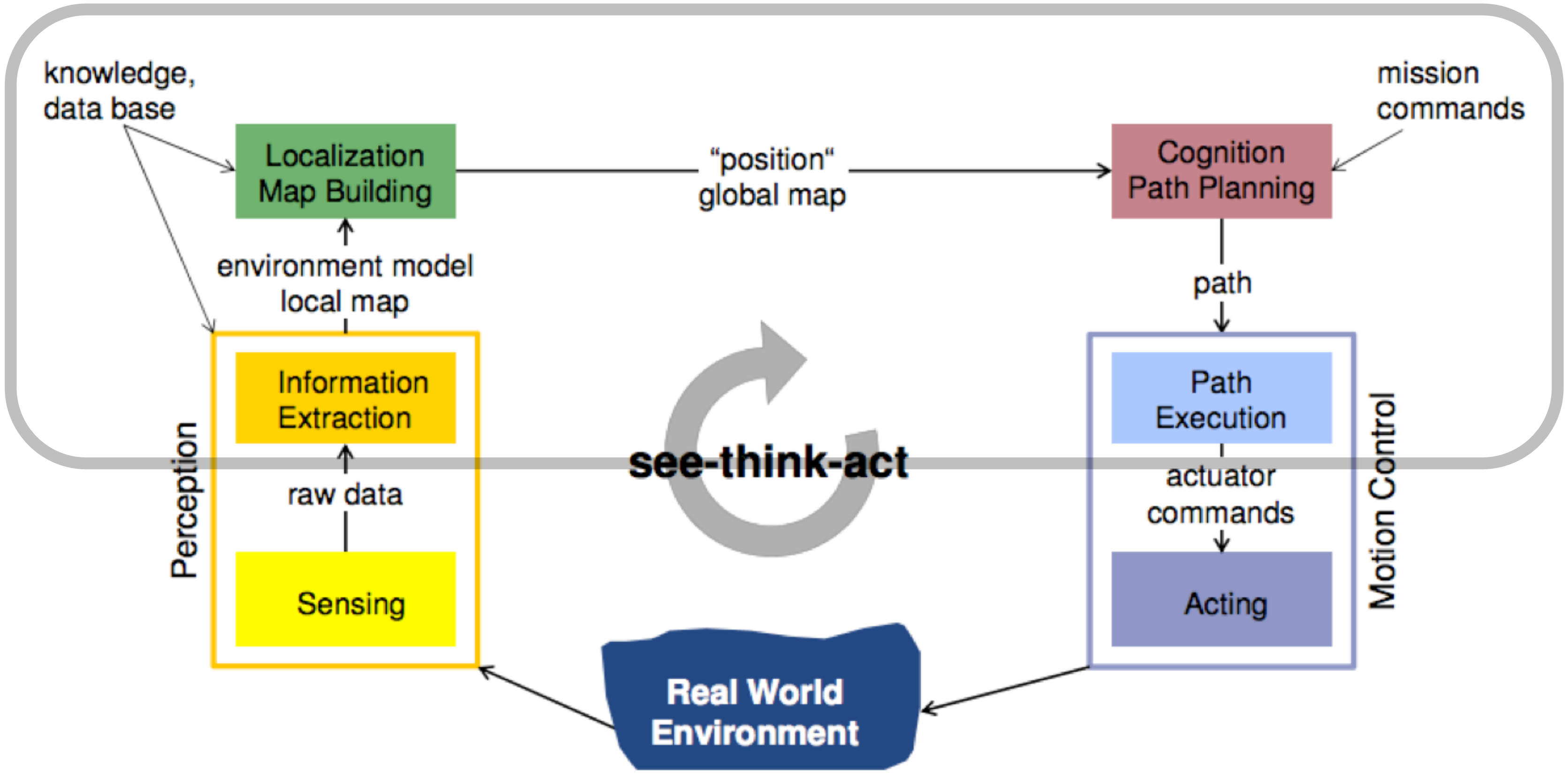
https://www.youtube.com/watch?v=0e3pEtECwLg

https://www.youtube.com/watch?v=0e3pEtECwLg

Creating Order from Chaos: Integrating HRI

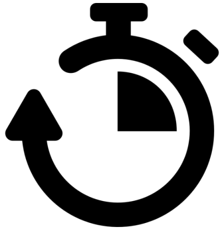


Robot Control in See-Think-Act Cycle

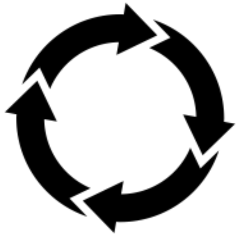


Integrating high-level (symbolic) planning with low-level control.

Software Engineering Robot Control



- **Real-time control**: supports event-based, reactive, and distributed interactions between sensors, motors and algorithms.

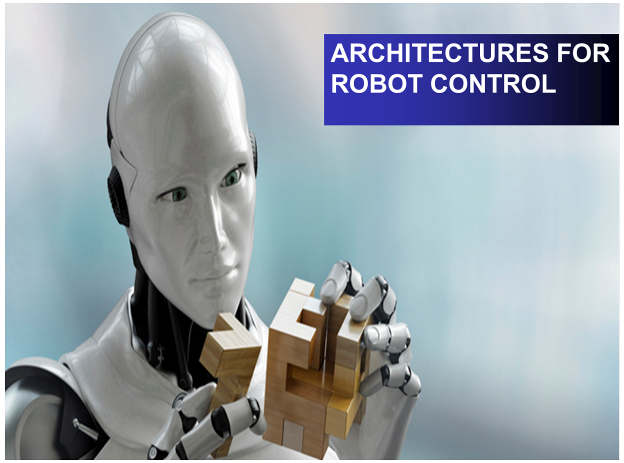


- **Reuse**: architecture is generic and abstracts from specific robot platforms for reuse.

- **Robustness**: ensures robust robot behavior and graceful degradation of task performance in case of failures.

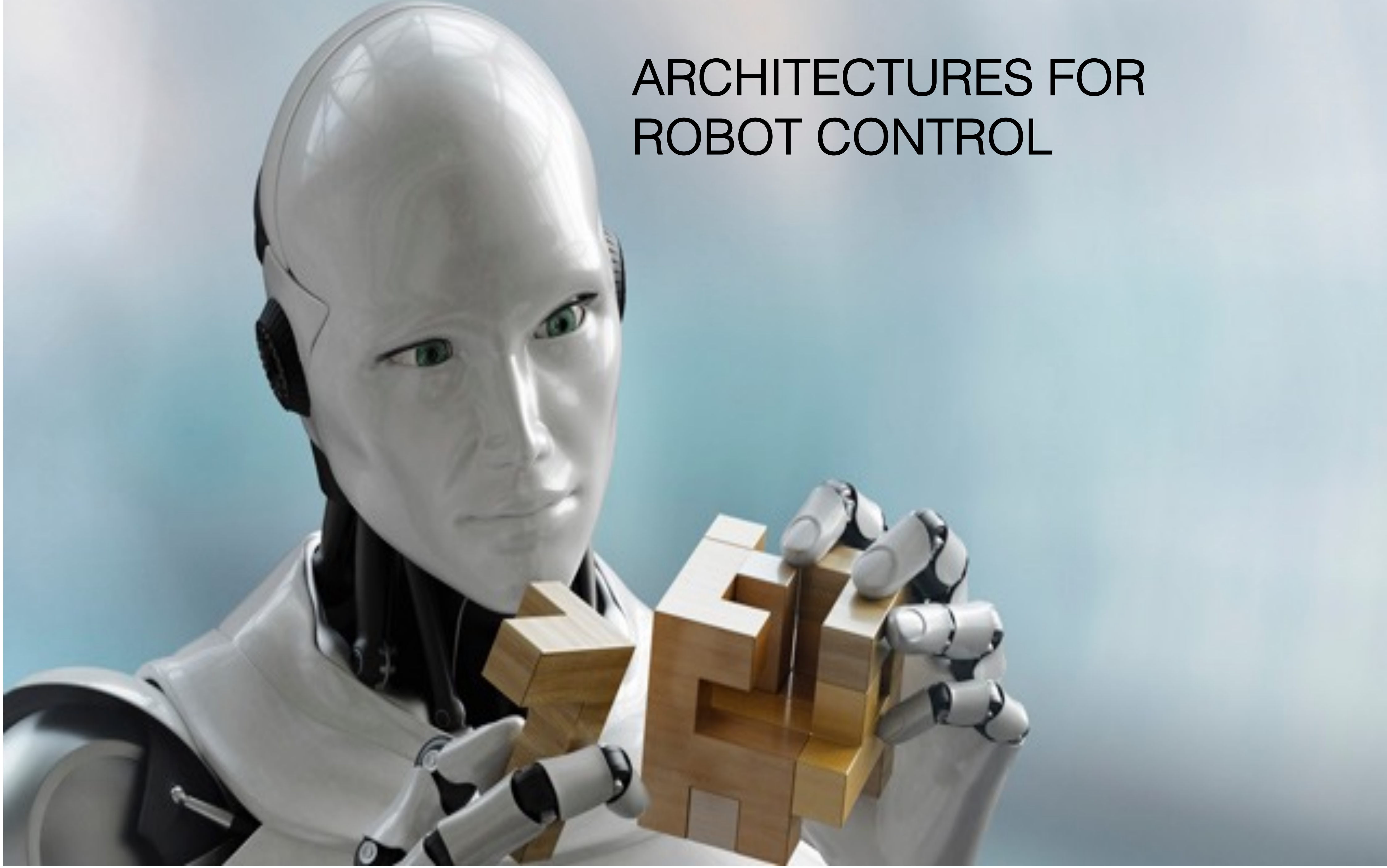
Outline


CHALLENGES FOR
ROBOT CONTROL

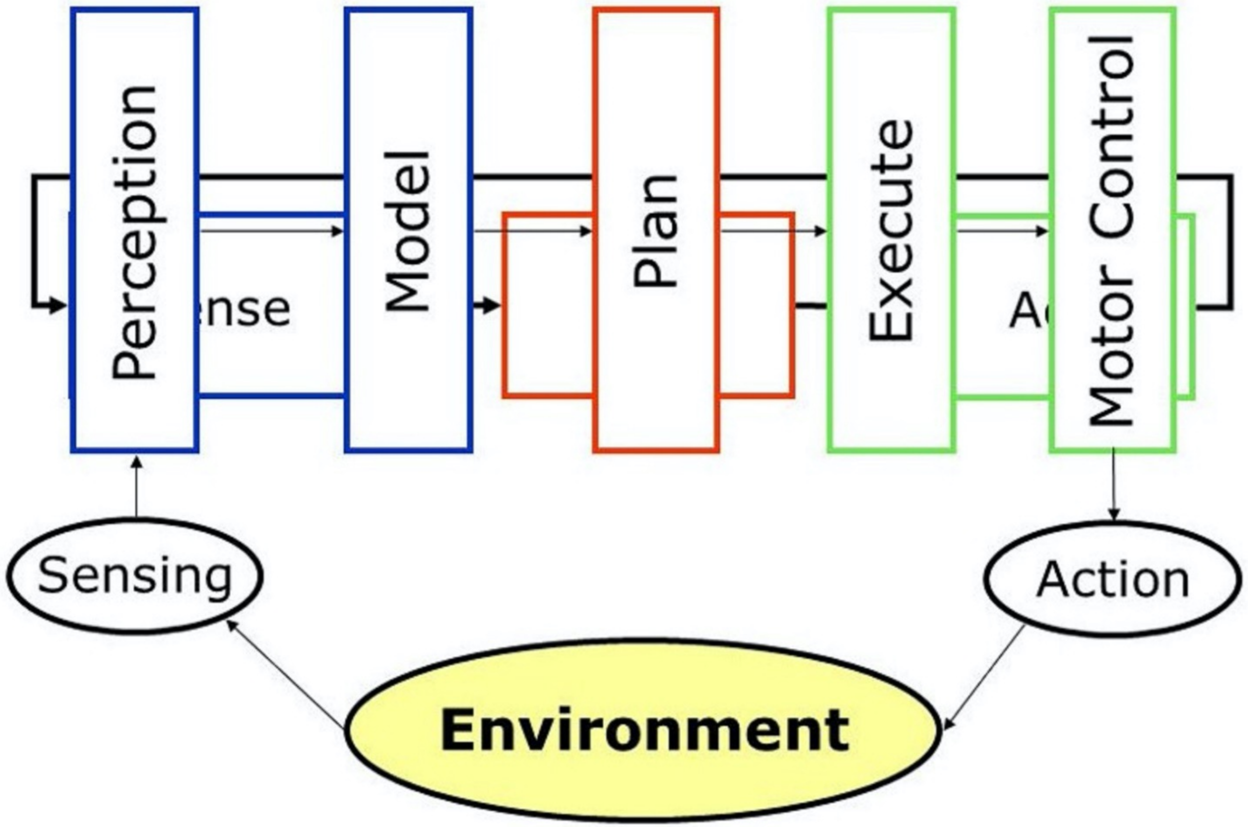

ARCHITECTURES FOR
ROBOT CONTROL


PERSISTENCE &
CLOUD ROBOTICS

ARCHITECTURES FOR ROBOT CONTROL

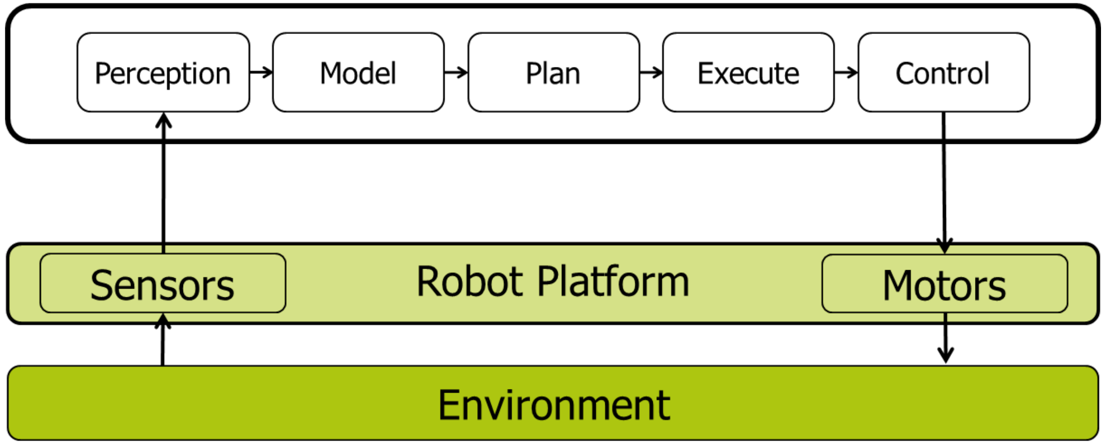


Sense-Plan-Act (SPA)



Classic version of a "pipeline" architecture.

Example: Shakey the Robot (1966-1972)



Classic SPA Architecture: Benefits & Issues

Benefit:

- Integrates symbolic and non-symbolic techniques.

Issues:

- Robot control *slow* due to “extensive deliberation”.
- Not very *robust* (no monitoring of task execution).

Artificial Intelligence 47 (1991) 139–159
Elsevier

Behavior-based robotics

Intelligence without representation*

Rodney A. Brooks

MIT Artificial Intelligence Laboratory, 545 Technology Square, Rm. 836, Cambridge, MA 02139, USA

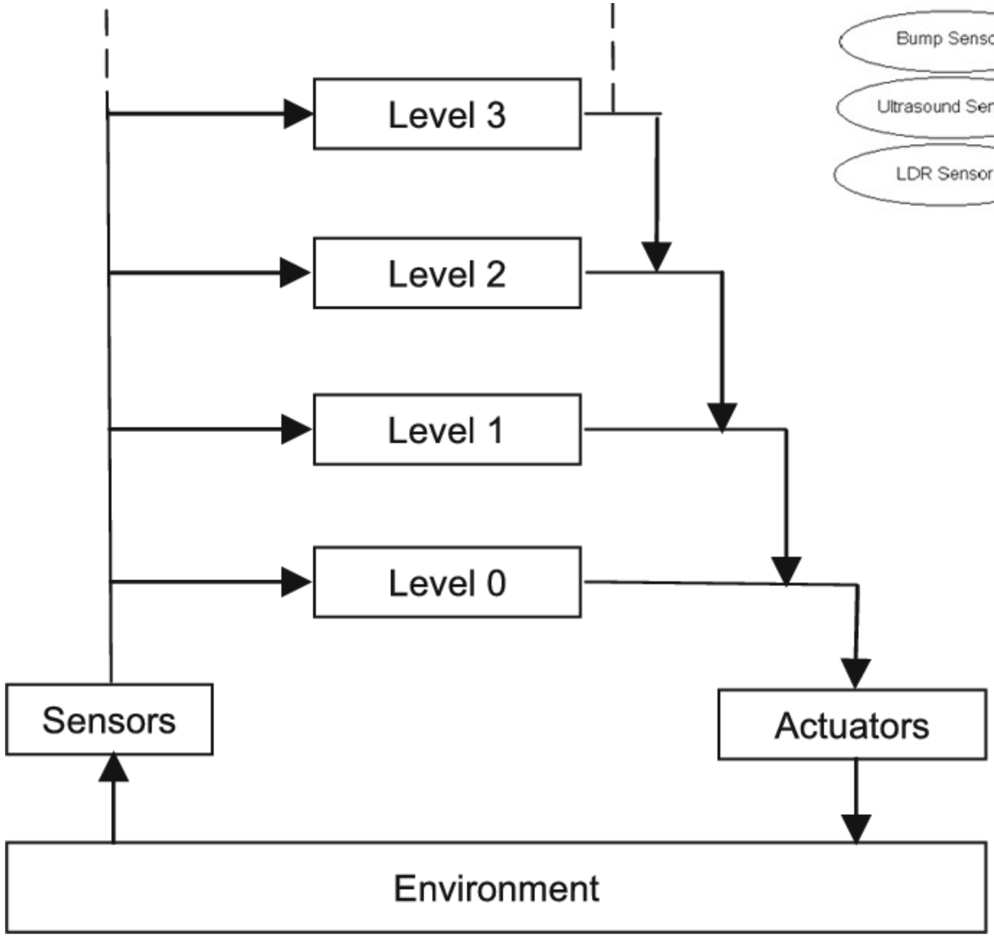
Received September 1987

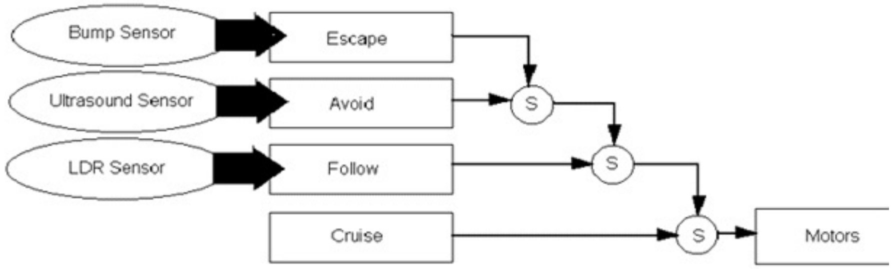
Abstract

Brooks, R.A., Intelligence without representation, Artificial Intelligence 47 (1991) 139–159.

Artificial intelligence research has foundered on the issue of representation. When intelligence is approached in an incremental manner, with strict reliance on interfacing to the real world through perception and action, reliance on representation disappears. In this paper we outline our approach to incrementally building complete intelligent Creatures. The fundamental decomposition of the intelligent system is not into independent information processing units which must interface with each other via representations. Instead, the intelligent system is decomposed into independent and parallel activity producers which all interface directly to the world through perception and action, rather than interface to each other particularly much. The notions of central and peripheral systems evaporate—everything is both central and peripheral. Based on these principles we have built a very successful series of mobile robots which operate without supervision as Creatures in standard office environments.

Subsumption Architecture (1985)

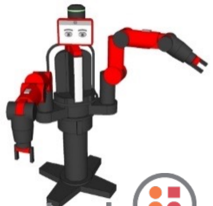




Rodney Brooks

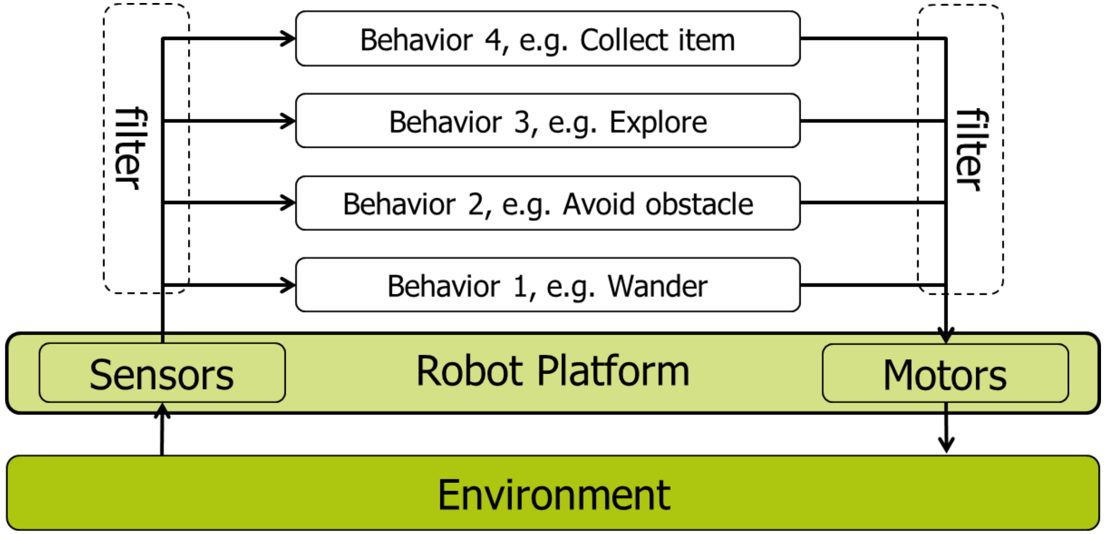
Robot Genghis

iRobot



rethink robotics

Robot Insects (1990)



Subsumption Architecture: Benefits & Issues

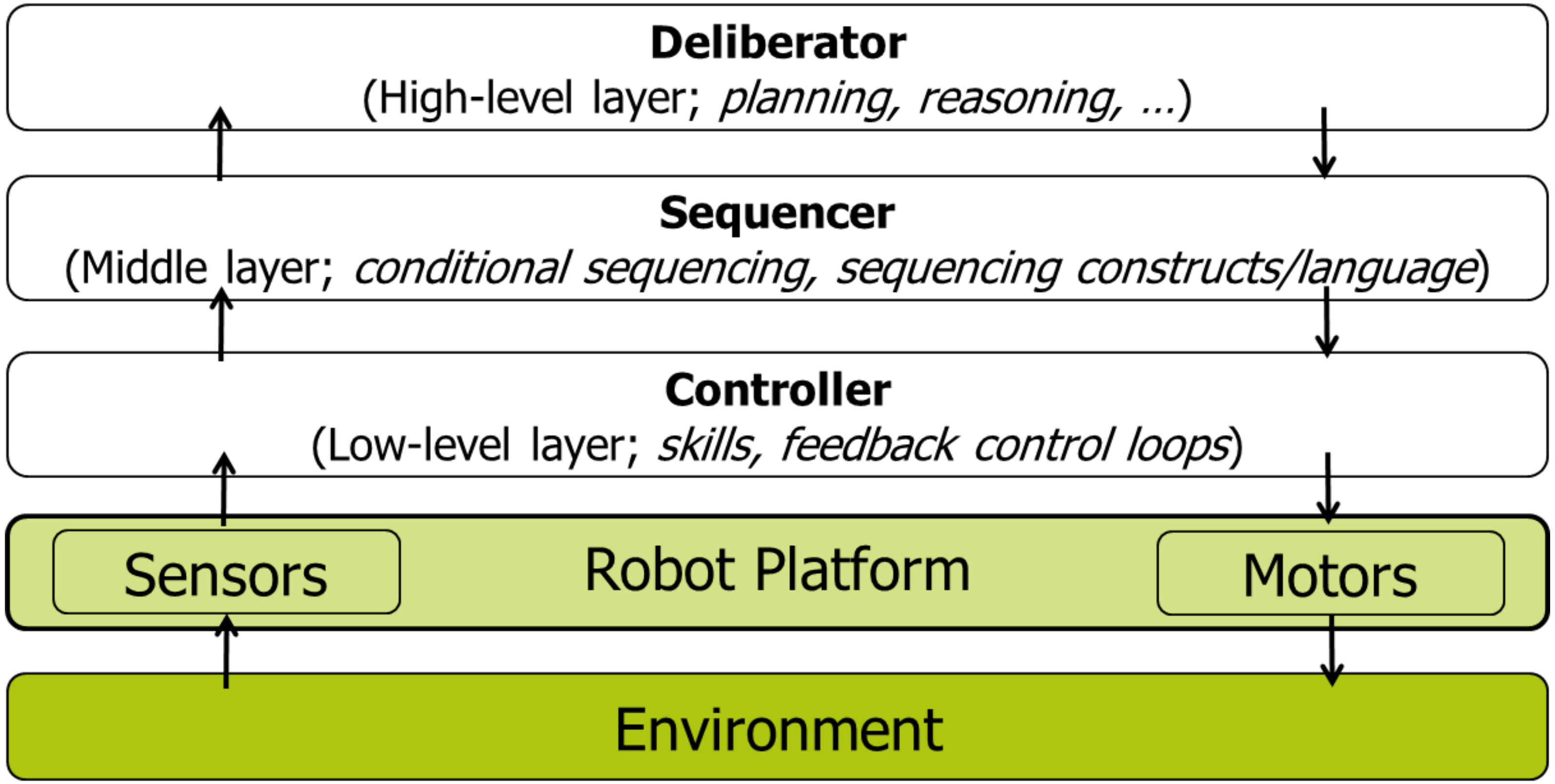
Benefits:

- Fast due to focus on behavior and tight sensor-behavior coupling (gave rise to behavior-based paradigm).
- Reactive, able to handle dynamic world due to constant sensing of the world.

Issues:

- difficult to compose behaviors to achieve long-range goals.
- almost impossible to optimize robot behavior.

3T or Layered Architectures



Classic examples: SSS (Connell 1991), ATLANTIS (Gat 1991), 3T (Bonasso 1991)

A Functional Perspective

3T

- (highest) *deliberative layer* responsible for task-planning and achieving long-term goals of the robot within resource constraints.

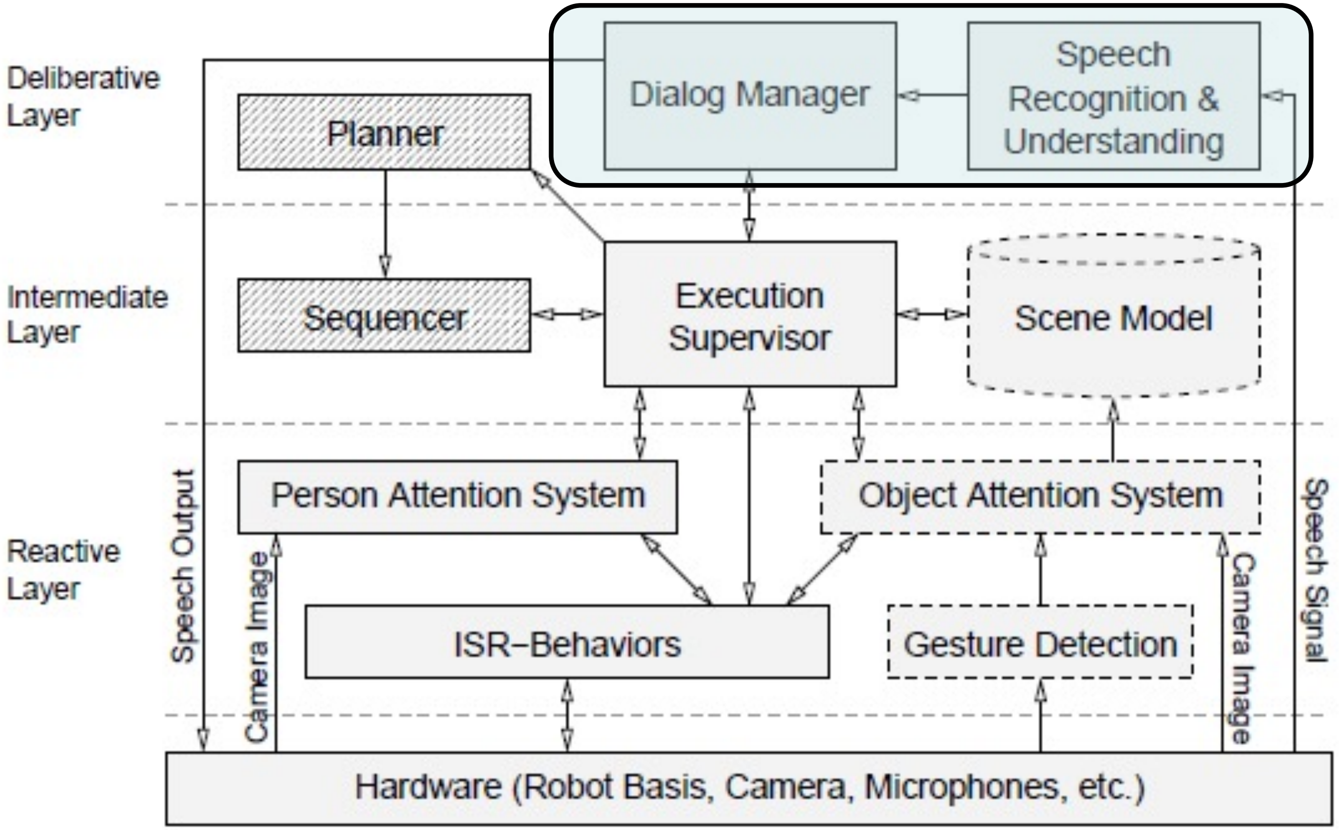
2T

- (middle) *executive or sequencer layer* responsible for choosing the current behaviors of the robot to achieve a task.

1T

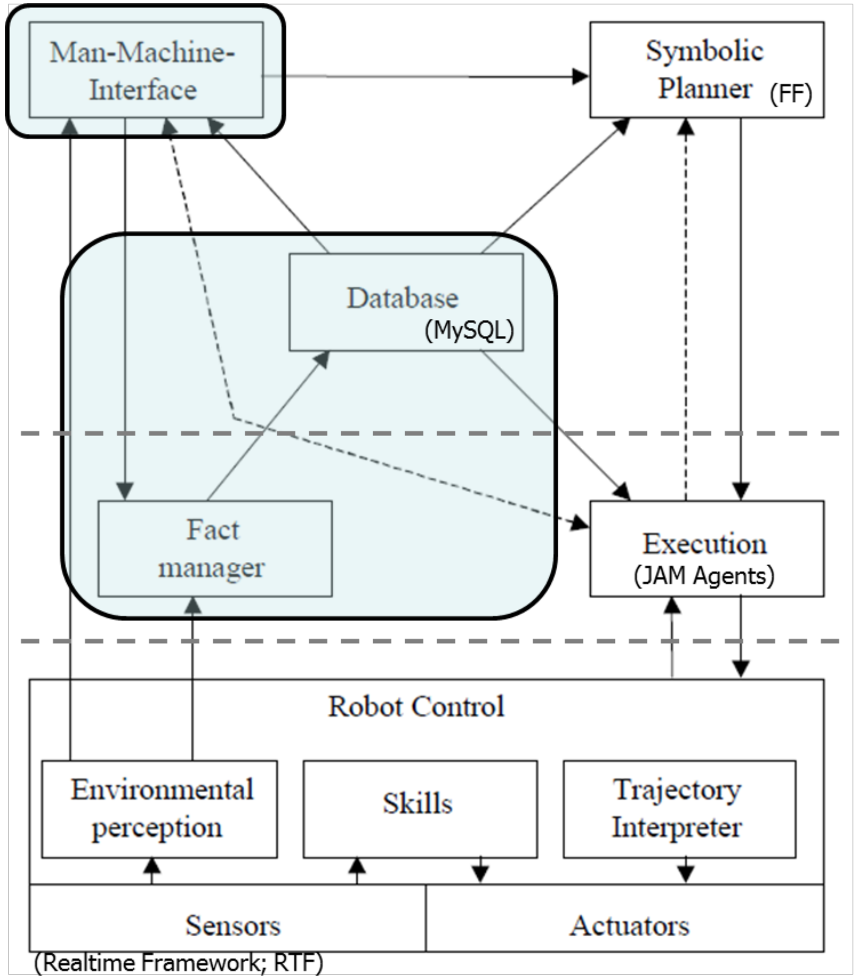
- (lowest) *behavioral control or skills layer* responsible for controlling *sensors* and *actuators*.

BIRON (2004)

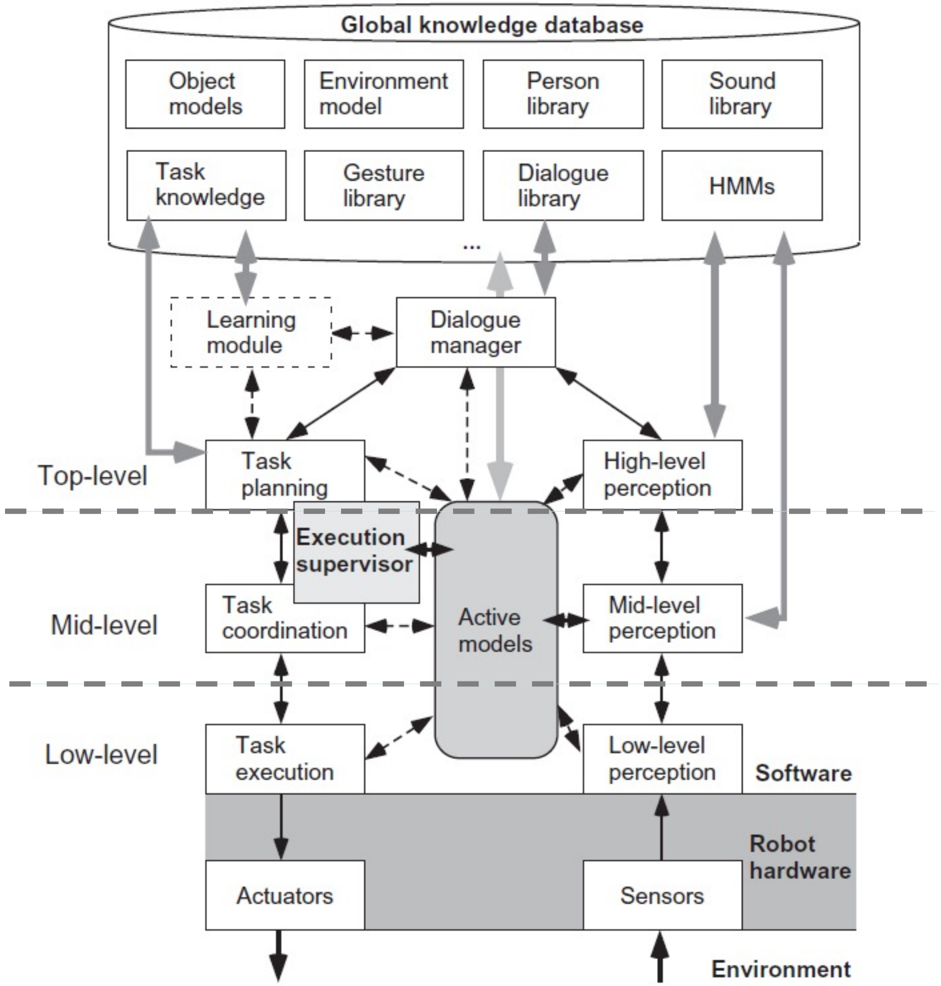


The Bielefeld Robot Companion

Care-O-bot II/3



Armar (Univ. of Karlsruhe)



3T Architecture: Benefits & Issues

Benefits:

- Rich architecture, with different levels of abstraction and clear “roles”: planning, execution (control and monitoring), and basic control layer (behaviors, …)

Issues:

- Complex: many ways to instantiate 3T, what is best?
- How many layers: >3 layers? Perhaps 2 layers?

→ Where to plan? Path planning at middle or highest layer?

Social Interaction Architecture

Support for:

- Express emotions
- Perceive emotions
- Communicate with high-level dialog
- Model other agents
- Use social cues naturally
- Explain behavior
- Learn from interaction
- …

Finding the Middle Ground

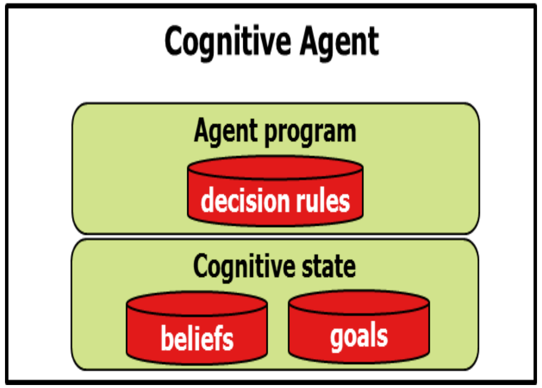
Options available for developing an interaction architecture

1

General-purpose

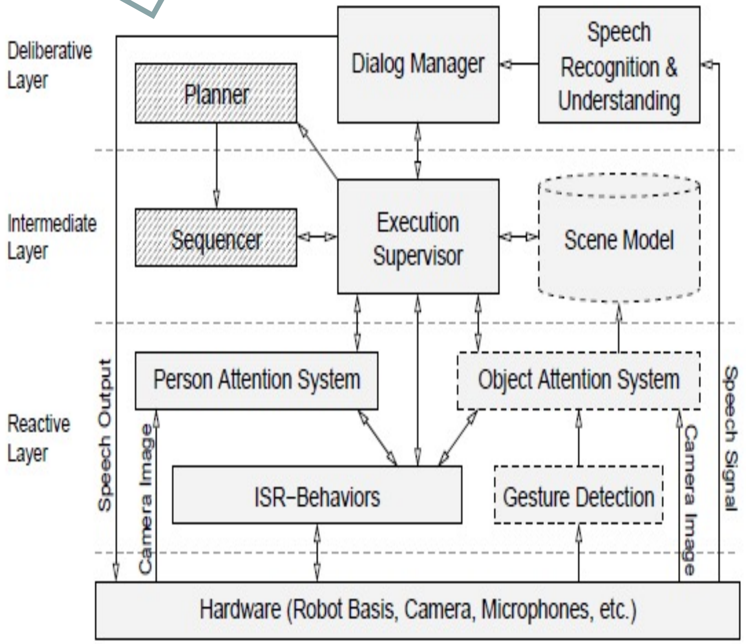
3

Hindriks (2009)



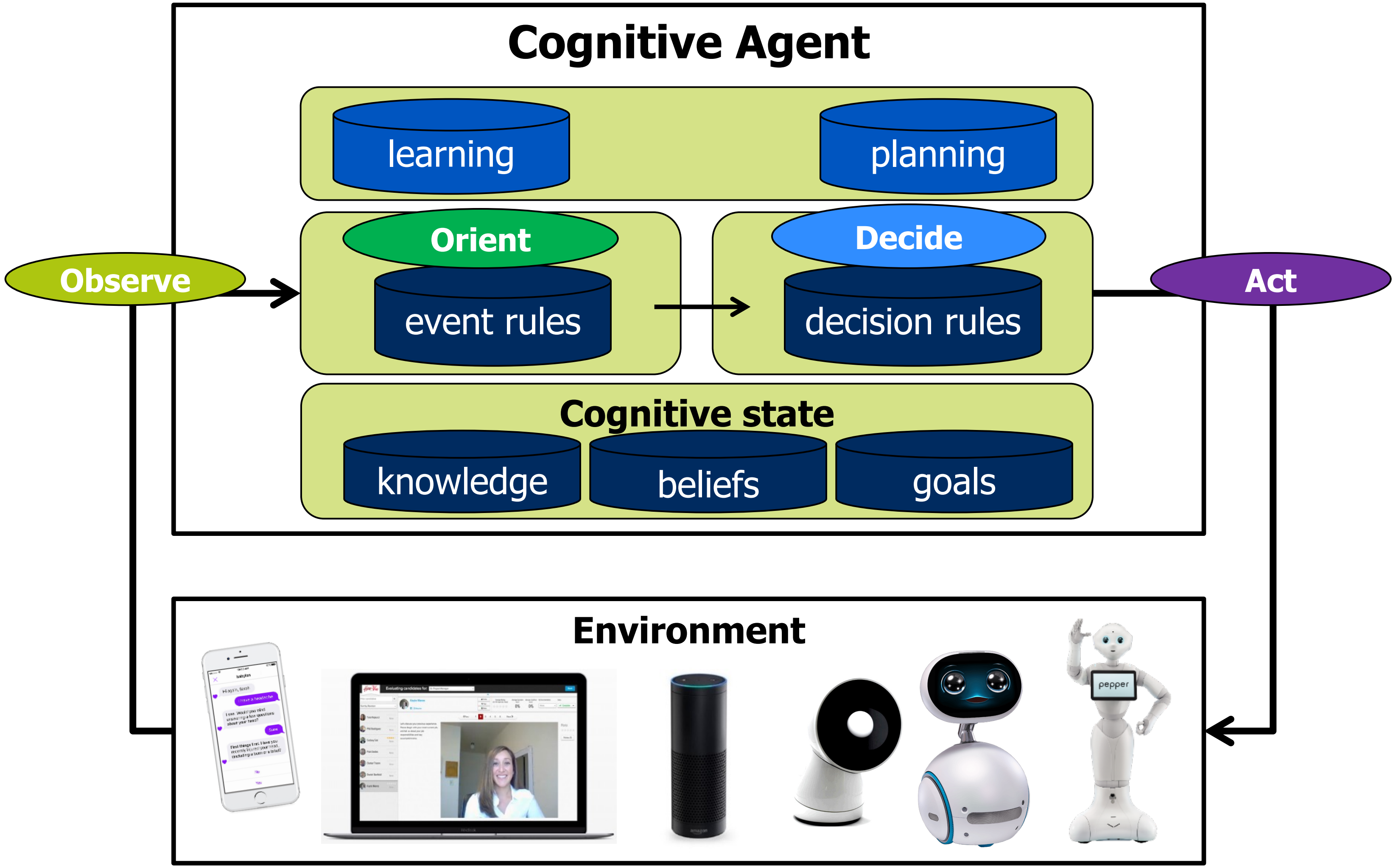
DSL for declarative decision-making

2



Typical 3-layer architecture

Cognitive Agents



Example Cognitive State

```
knowledge{
  % TOPICS-SF order
  order([a, b, c, d1, d2, …]).
  next(OrderEl) :- done(Items), order(All),
    append(Items, [OrderEl | _], All).
  …
}
beliefs{
  dialogStructure([]).
  done([a, b]).
  % dialog(TopicLabel, OrderEl, DialogEl, Txt, Pars)
  …
  dialog(topics-sf, c, question, "Heeft u klachten over uw
    geheugen?", ['ja', 'nee']).
  …
}
goals{
  dialogStructure([topics-sf, pwi, rs]).
}
```

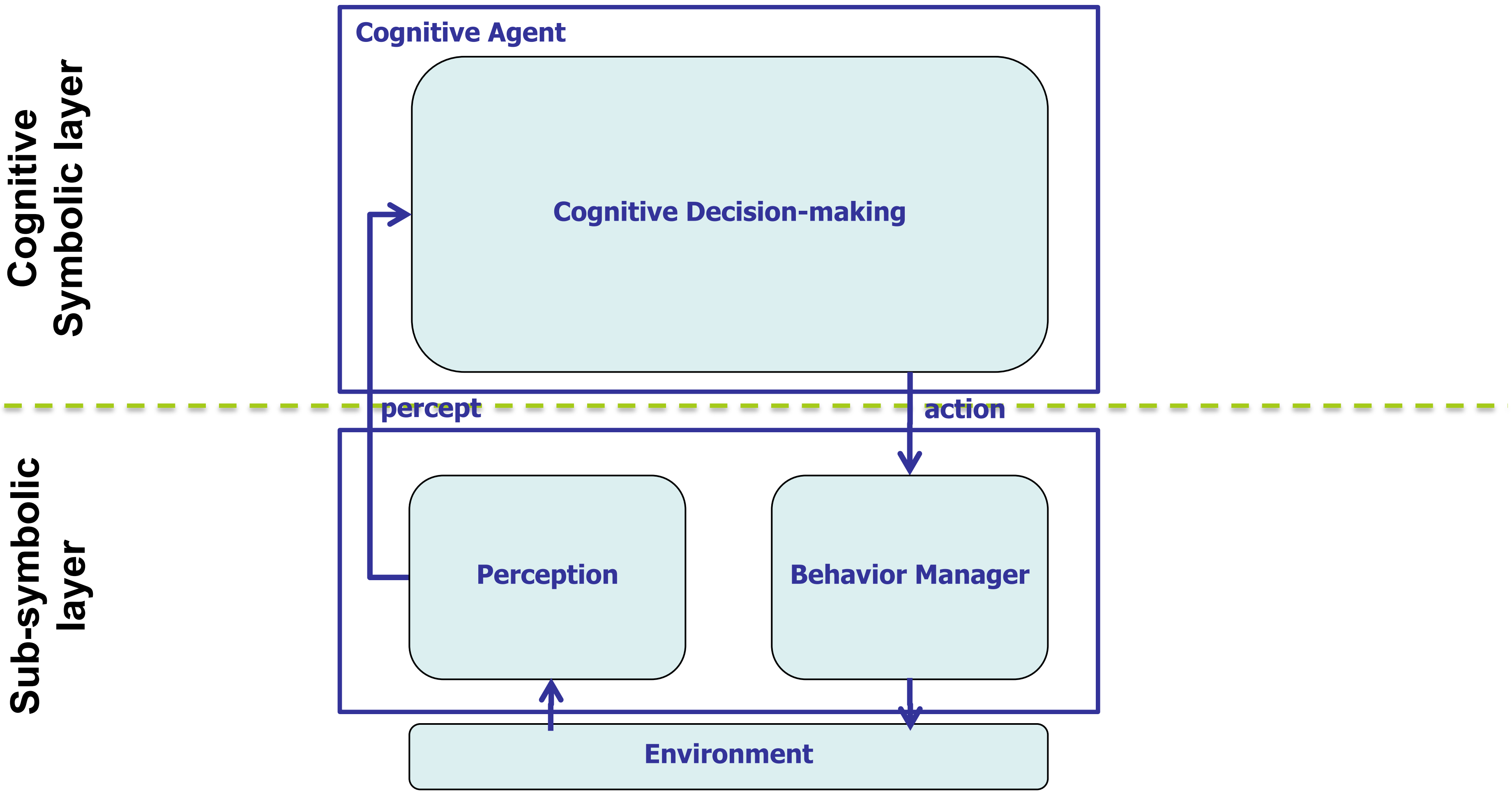
Domain knowledge (static), beliefs, and goals represented in e.g., Prolog, OWL.

Example Agent Program

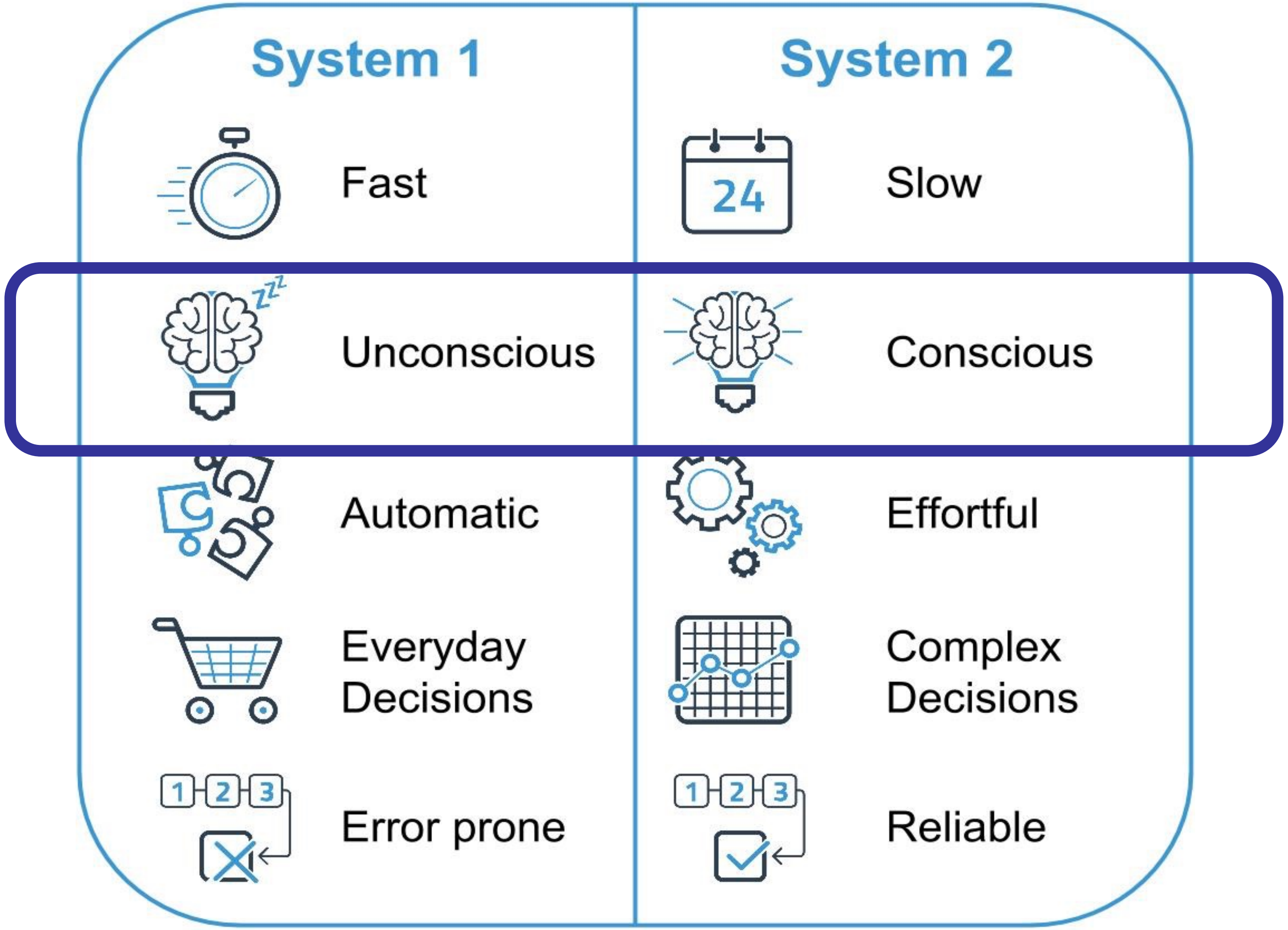
```
init module{
  …
  actionspec{
    define ask(Text, Options)
        with pre{ speechEnabled } post{ true }
  }
}

event module{
  …
  if percept(speechEnabled) then insert(speechEnabled).
  …
}

main module{
  …
 if bel(next(OrderEl), dialog(topics-sf, OrderEl,
    question, Txt, Pars)) then ask(Text, Options).
  …
}
```

More information at https://goalapl.atlassian.net/wiki/spaces/GOAL/

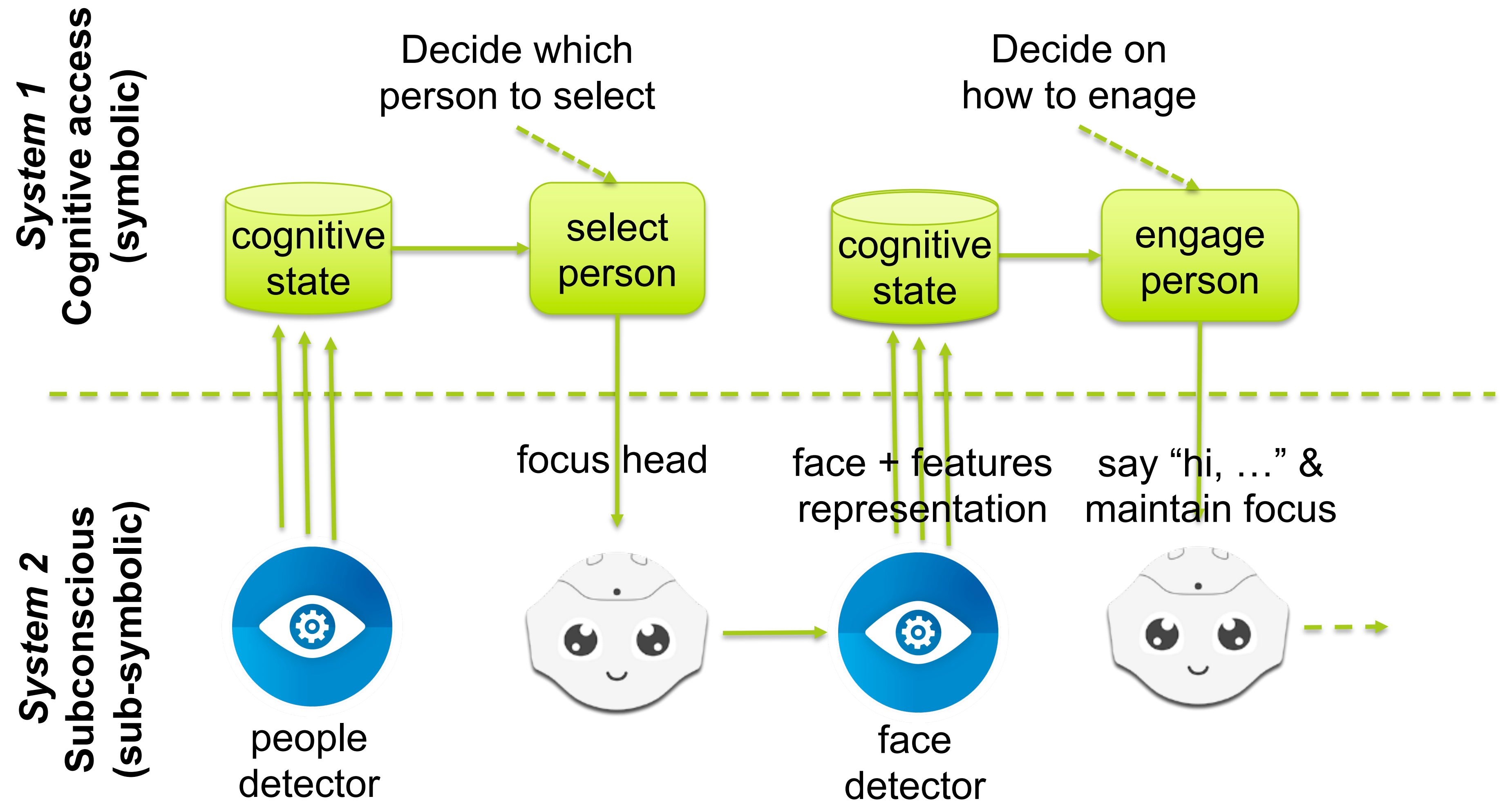
2-Layered Interaction Architecture



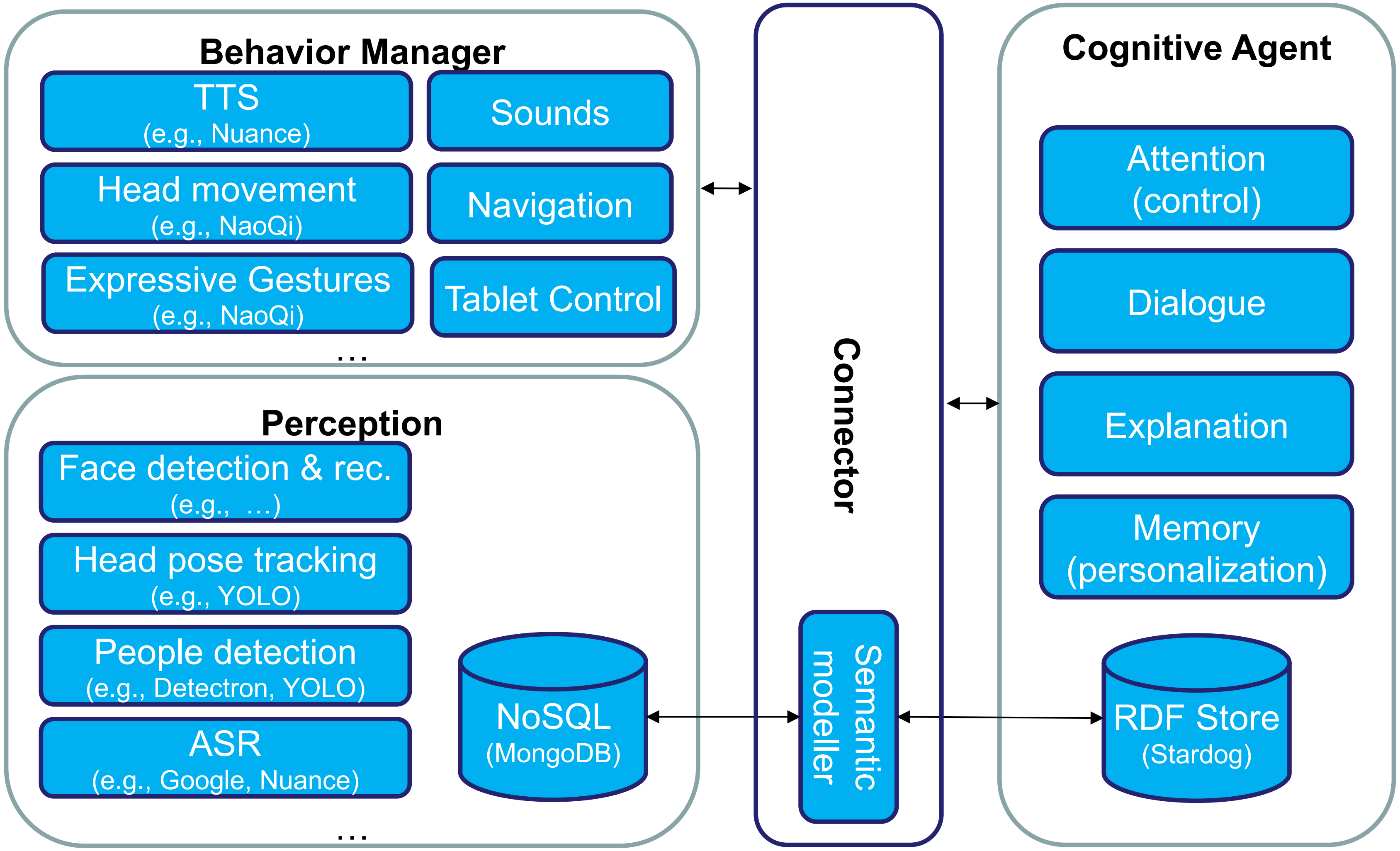
System 1 & 2 Architecture Design



Example: Attention/Engagement

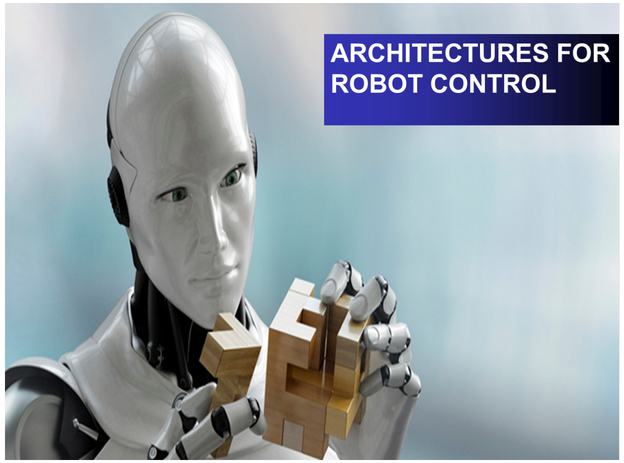


Technical Architecture



Outline


CHALLENGES FOR
ROBOT CONTROL


ARCHITECTURES FOR
ROBOT CONTROL

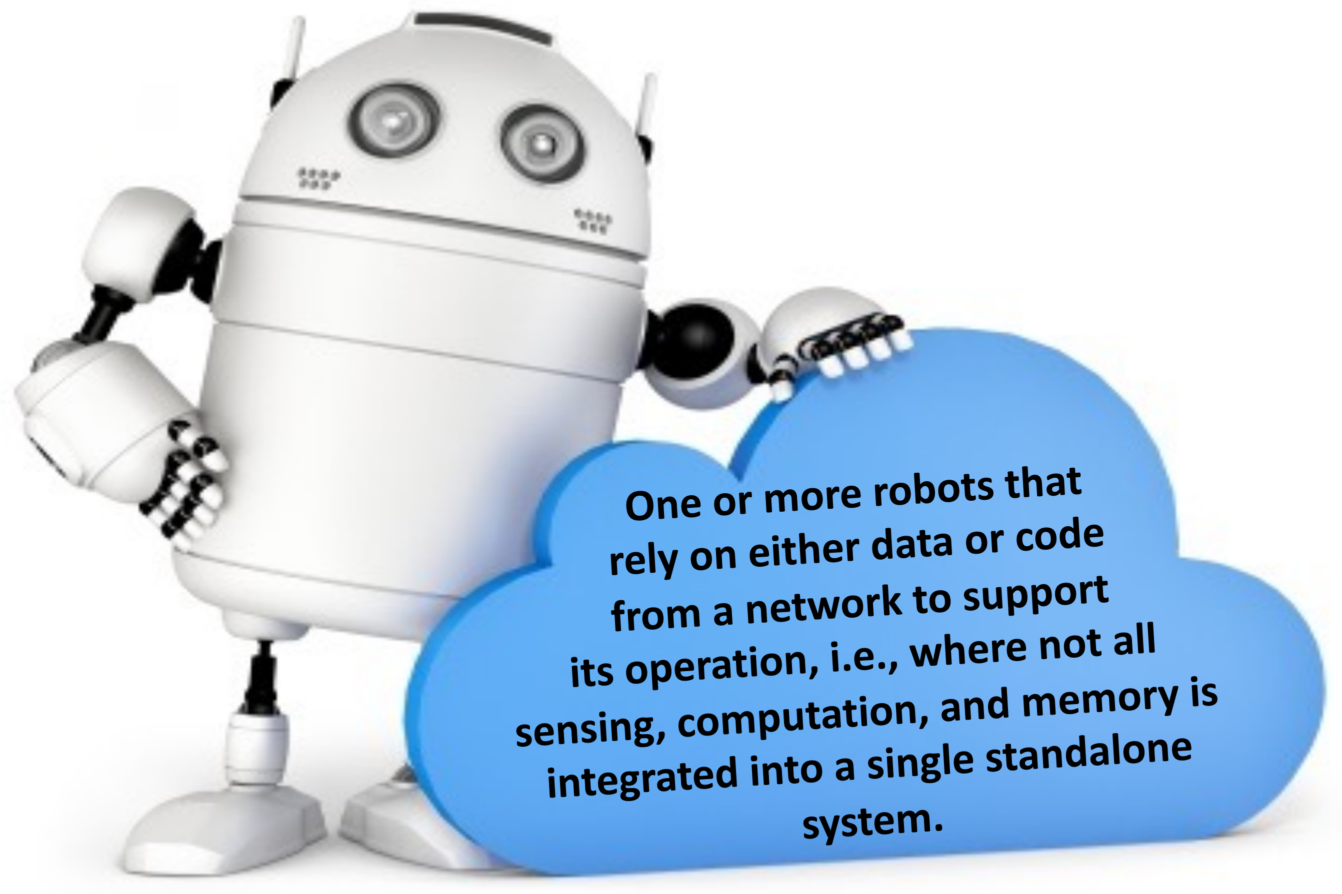

PERSISTENCE &
CLOUD ROBOTICS

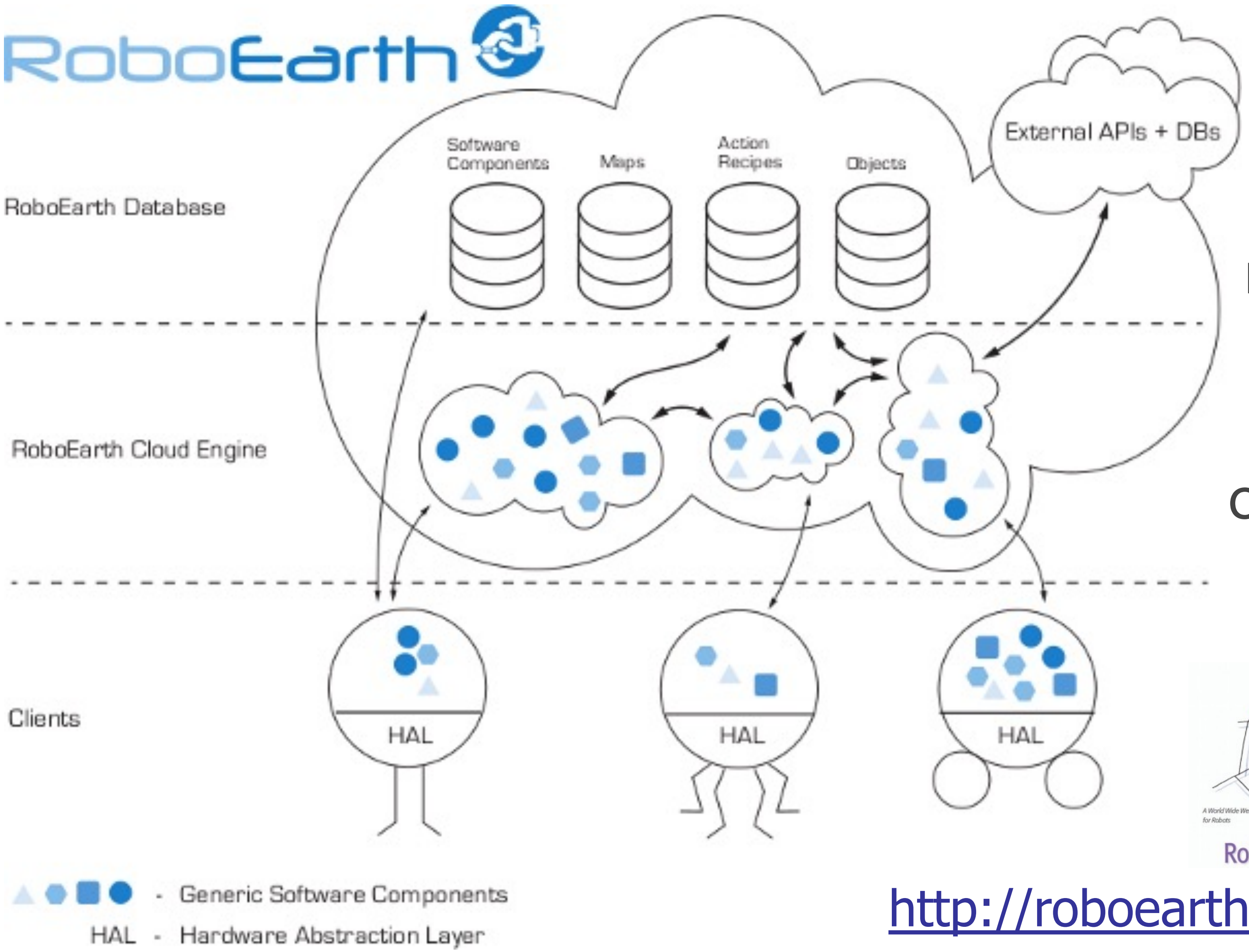
PERSISTENCE & CLOUD ROBOTICS

Persistence

- Extend architecture and integrate database technology to ensure required persistence.
- Useful to be able to store “raw” data (images, video, etc.), and different types of data in a document store such as MongoDB.
- For high-level (more abstract) knowledge representation, use e.g. rdf triple stores such as Stardog.

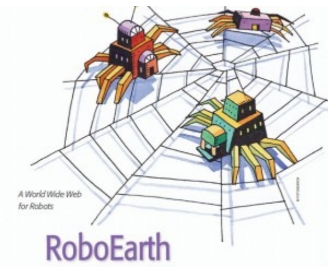
Cloud Robotics





Store and Share Information

Offload Computation



http://roboearth.ethz.ch/

https://www.youtube.com/watch?list=PL8Jt2DDNgBLo2-XxRStWTlMhwGcwdanfh&v=mgPQevfTWP8

Cloud Robotics: Benefits

- **Data** in the Cloud: Access to online resources (big data).

- **Computing** in the Cloud: Access to online services (e.g., navigation, learning).

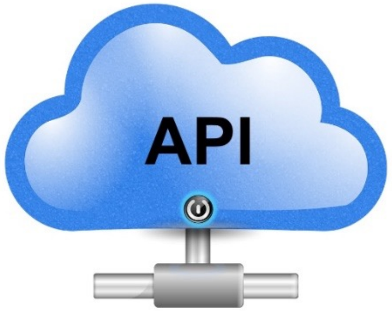
- **Sharing** in the Cloud: Robots sharing situated knowledge, performance, and outcomes.

- **Easy Access** via the Cloud: control and human collaboration via web.

Cloud Robotics' Challenges

- **Network unreliability**: how to handle missing packages, low-latency, unavailability?

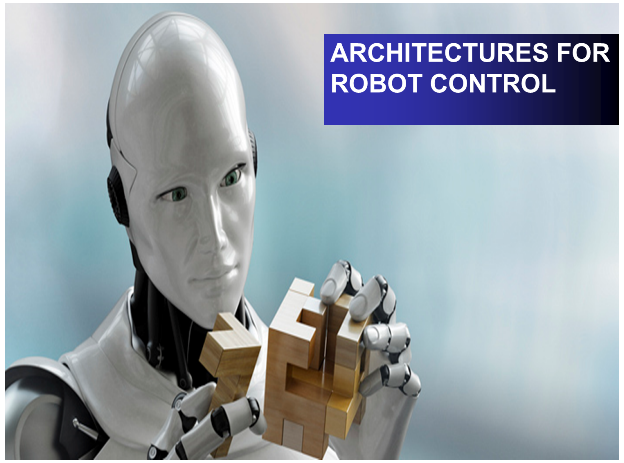
- **Privacy and security**: how to handle privacy concerns, data ownership and security?



- **Service integration**: how to integrate and connect with other (legacy) services?

Summary


CHALLENGES FOR
ROBOT CONTROL

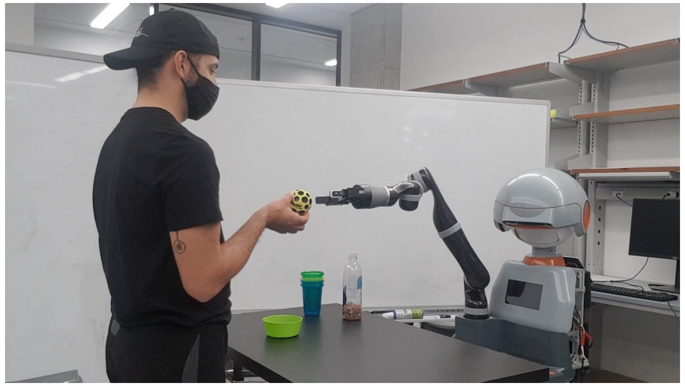

ARCHITECTURES FOR
ROBOT CONTROL


PERSISTENCE &
CLOUD ROBOTICS

Case study: Interactive Task Learning (ITL)

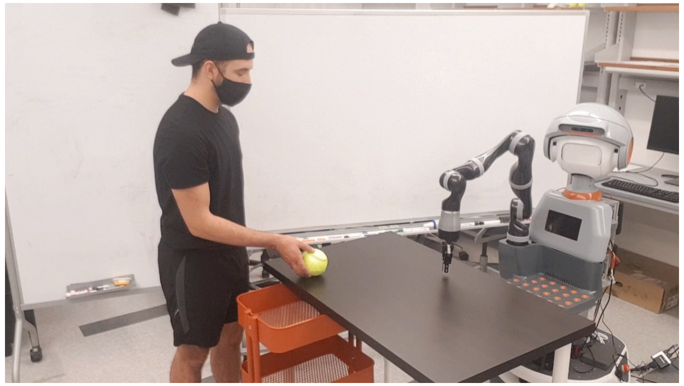
Interactive task learning is the problem of learning a task purely from (social) interaction with human teachers, including the goal (or reward function), an optimal plan or policy, human preferences, etc.

Examples of tasks for ITL



Handover

Pouring



Ball shooting

Brushstrokes

Creative painting

Expressive motion

Discussion: what components are required to make ITL happen?

- Interpreting instructions
  - Pointing
  - Gaze
  - Demonstration (visual, kinesthetic, …)
  - Evaluative feedback
  - Corrective feedback
  - …
- Learning
  - Learning by imitation / from demonstrations
  - Interactive RL
  - Learning from action advice
  - Evolutionary approaches
- Planning
  - Active learning (asking questions)
  - Human-robot collaboration
  - Hierarchical skill acquisition