Formal Synthesis of Uncertainty Reduction Controllers

Marc Carwehl,
Calum Imrie, Thomas Vogel,
Genaína Rodrigues, Radu Calinescu, Lars Grunske

🥳 Imagine the following scenario

You are visiting a friend in a city you are unfamiliar with, and therefore you need to ensure you reach their party on time by finding your way from the city’s railway station to your friend’s house. You are aware that your phone will run out of battery if you constantly check that you're still on track on your phone. To preserve some of your phone's battery, you may have a notion of not needing to check your phone while following a straight path, but deciding to finally check it once you encounter an intersection. Additionally, you probably acknowledge environmental conditions, such as low visibility due to fog or rain, and adapt your behaviour accordingly. Maybe, you don't even rely solely on your phone to guide you, but look out for street names. If you are sophisticated, you may even save more of your phone's battery by remembering street names or landmarks you must traverse, thus ensuring you are still on track. 🍻

Inherent Uncertainties

- Unknown aspects in a specific situation
  - Location, direction
  - Sensor values
- Imperfect knowledge
- Problems when deciding what to do next

Human notion when dealing with uncertainty

- We can use resources to gain knowledge
- We take our surroundings into account
- We can trade off between mutiple objectives - all the time
- But: Our reasoning might not be perfect

So how do autonomous systems deal with uncertainty?

- Use Sensors/Services, reasoning
- Usually: If there's a sensor, we use it! We love data!
- Actions to reduce uncertainty, with cost!

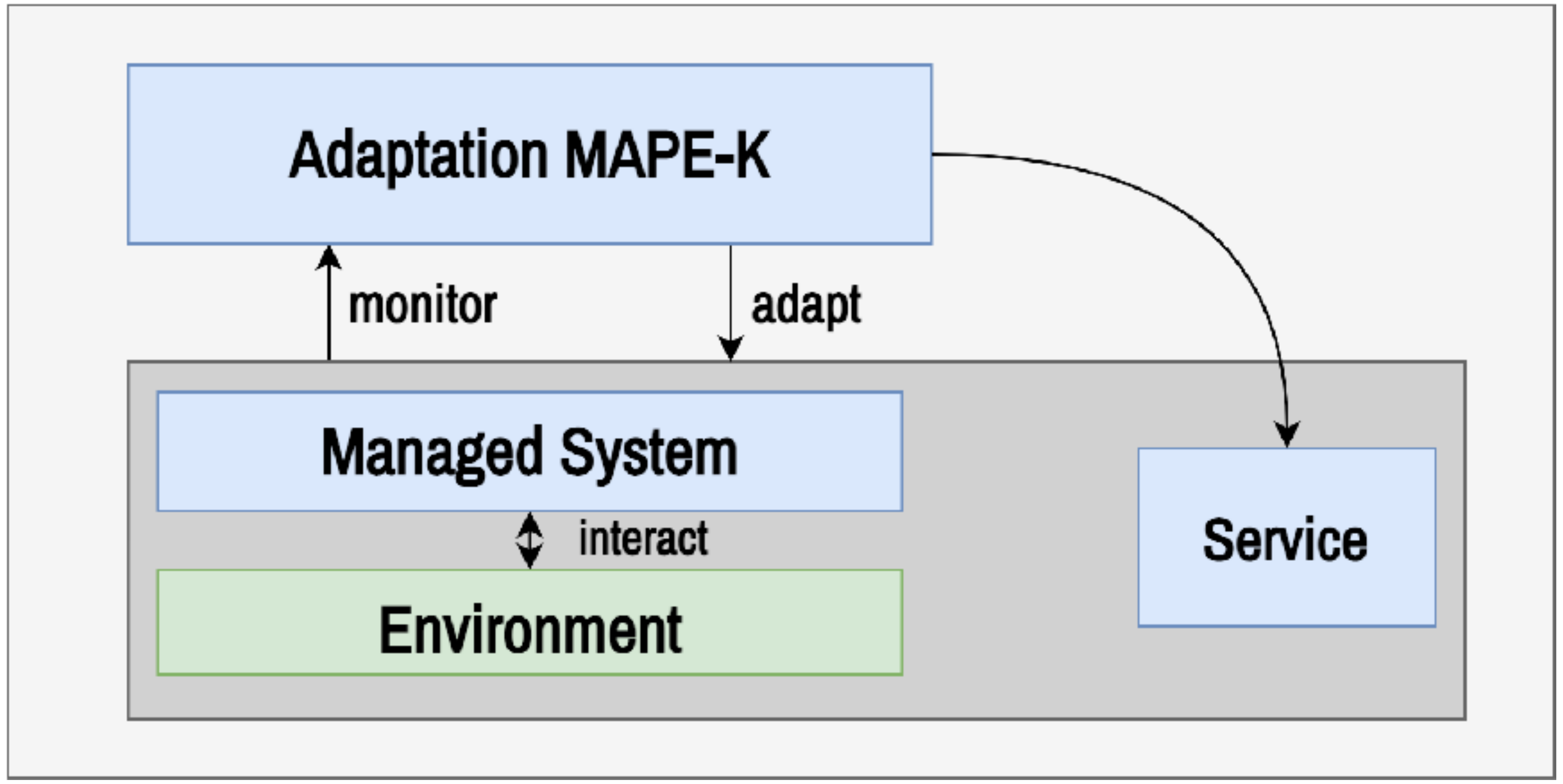
Uncertainty Reduction in Self-Adaptive Systems

Gabriel A. Moreno
gmoreno@sei.cmu.edu
Carnegie Mellon University
Software Engineering Institute
Pittsburgh, PA, USA

Javier Cámara
jcmoreno@cs.cmu.edu
Carnegie Mellon University
School of Computer Science
Pittsburgh, PA, USA

David Garlan
garlan@cs.cmu.edu
Carnegie Mellon University
School of Computer Science
Pittsburgh, PA, USA

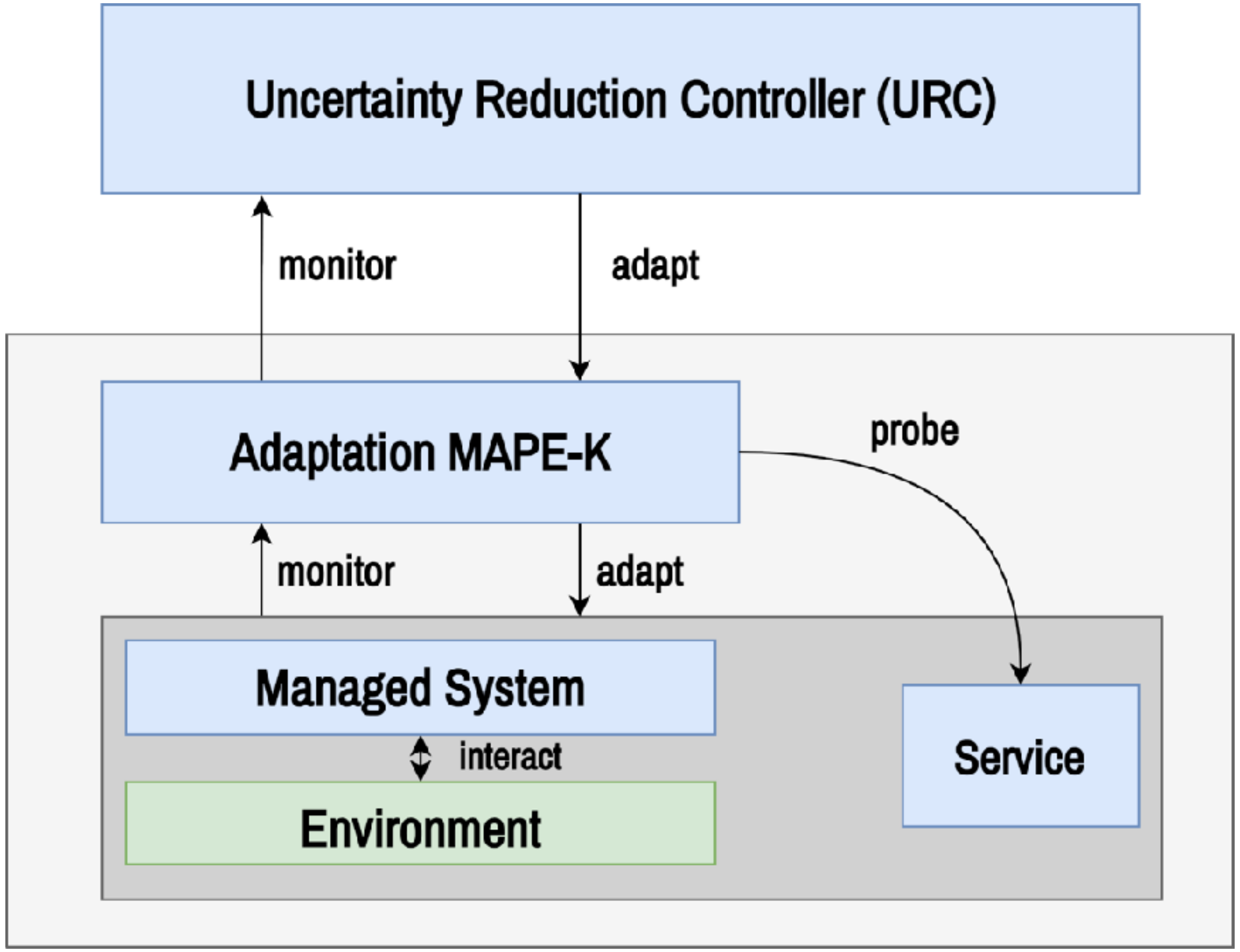
Mark Klein
mk@sei.cmu.edu
Carnegie Mellon University
Software Engineering Institute
Pittsburgh, PA, USA



Uncertainty Reduction Controllers (URC)

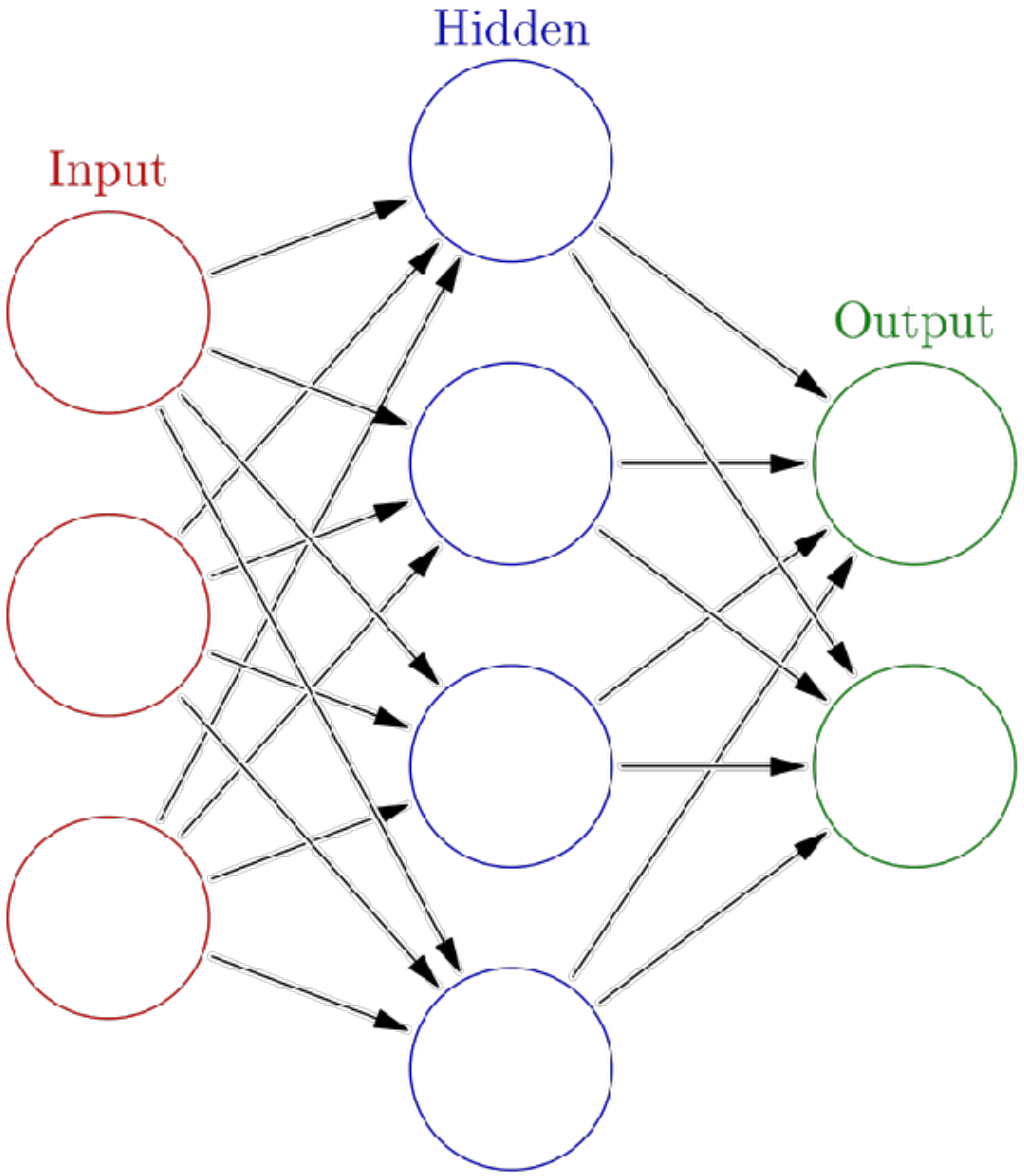
- Uncertainty Reduction as a first cla
  citizen

- Seperation of concerns



Controllers?!

- Decide which action to take in a particular scenario

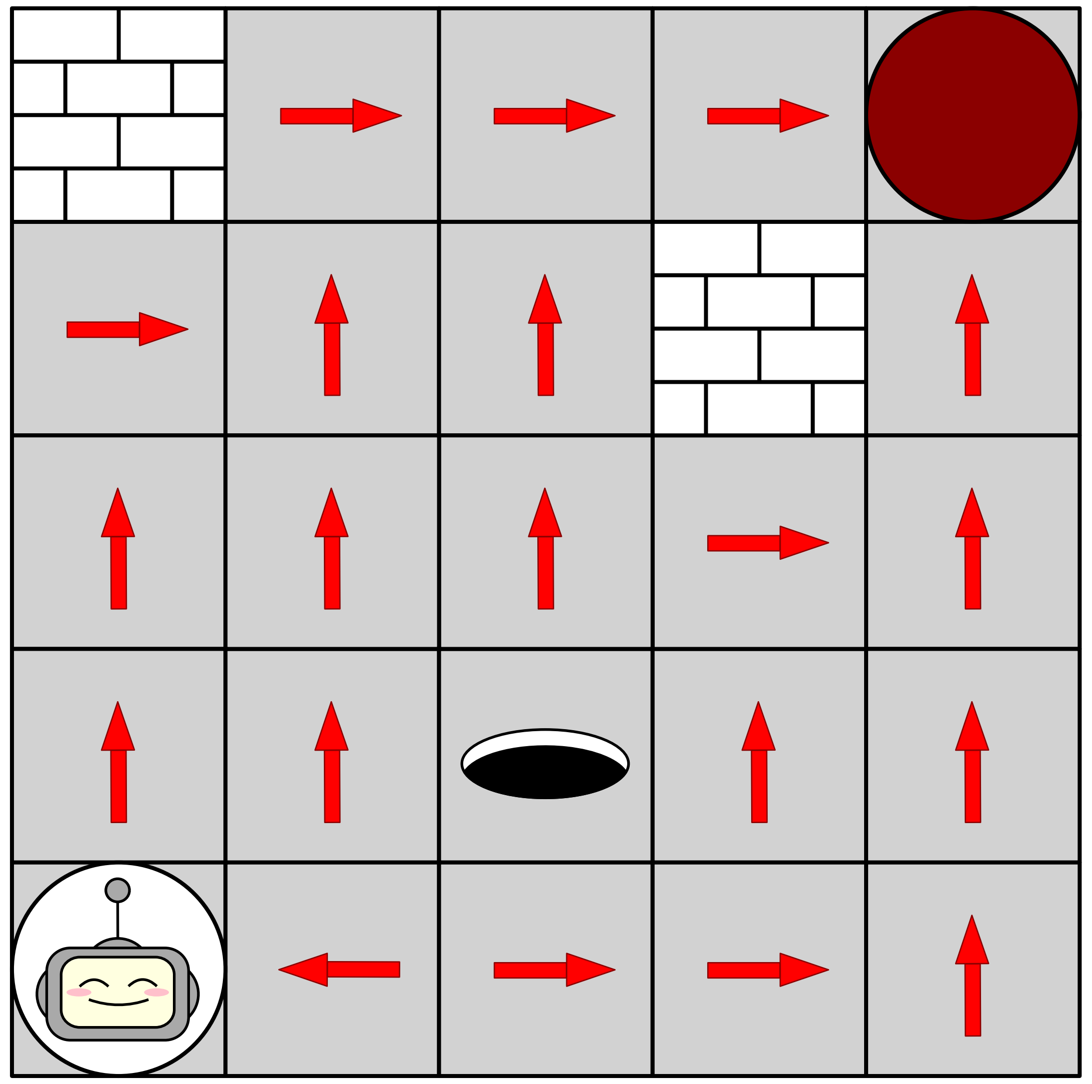


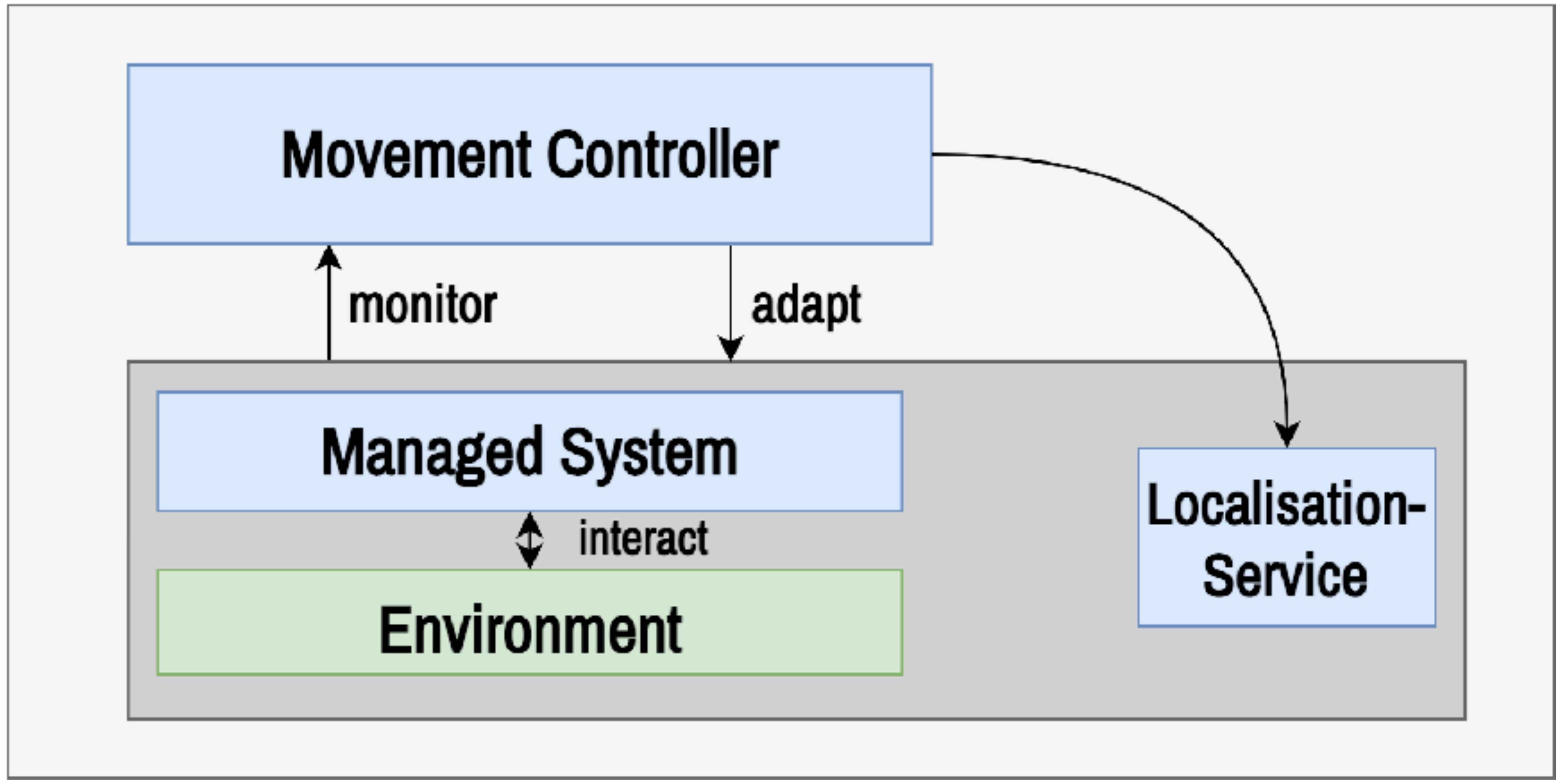
Controllers

- Mathematical function f: Scenarios -> Actions
- f_1(scenario) = A
- f_2(scenario) = B
- Compare f_1 and f_2

Discrete Robot Navigation

- Mission: reach destination
- Static Obstacles
- Position on the map
- Imperfect Move N, E, S, W
- Estimate about position
- Objectives: reach destination, low cost



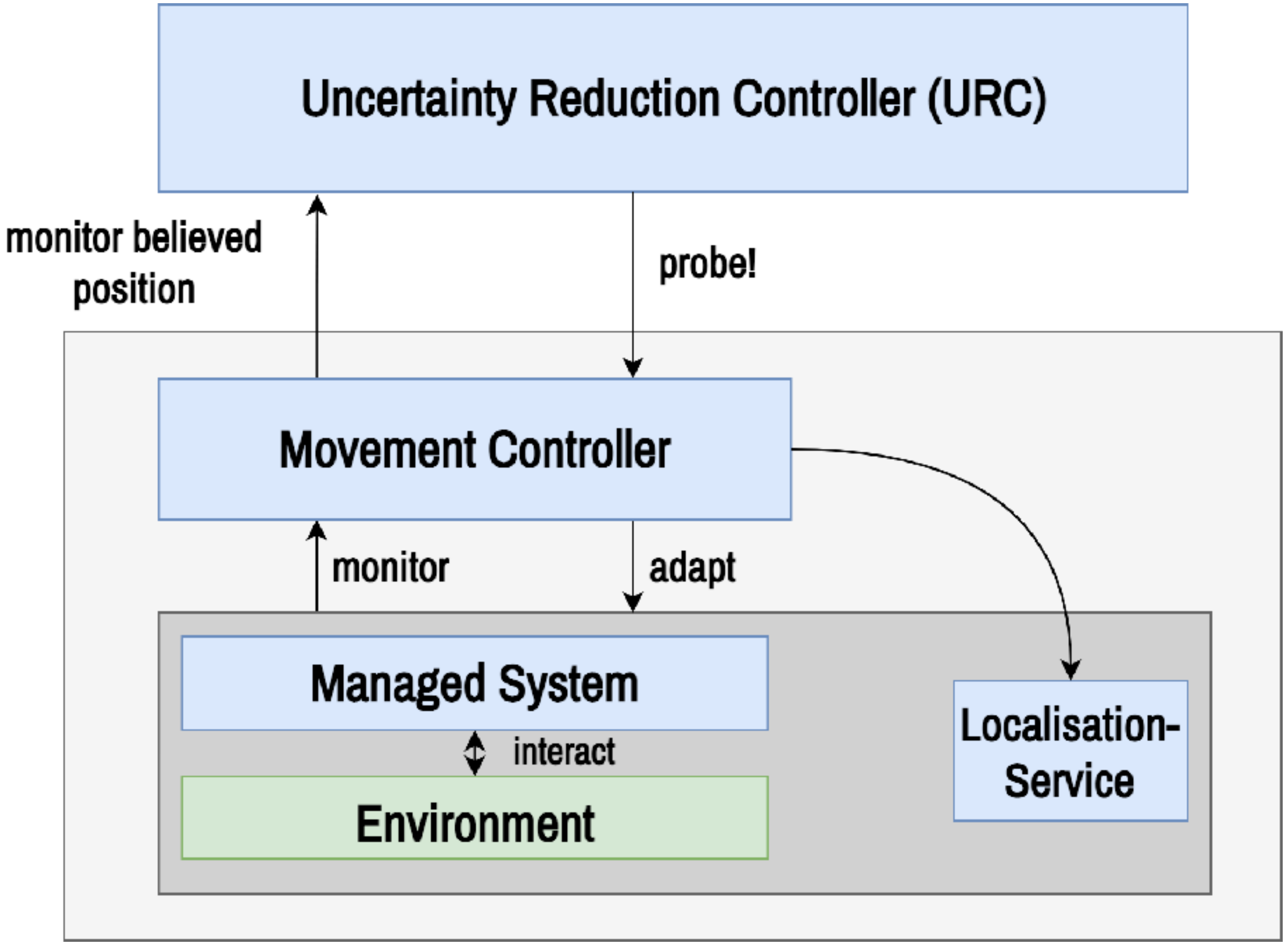


URC for the Robot

- When should the robot use localisation-service?
  Which frequency?

- Based on estimated position

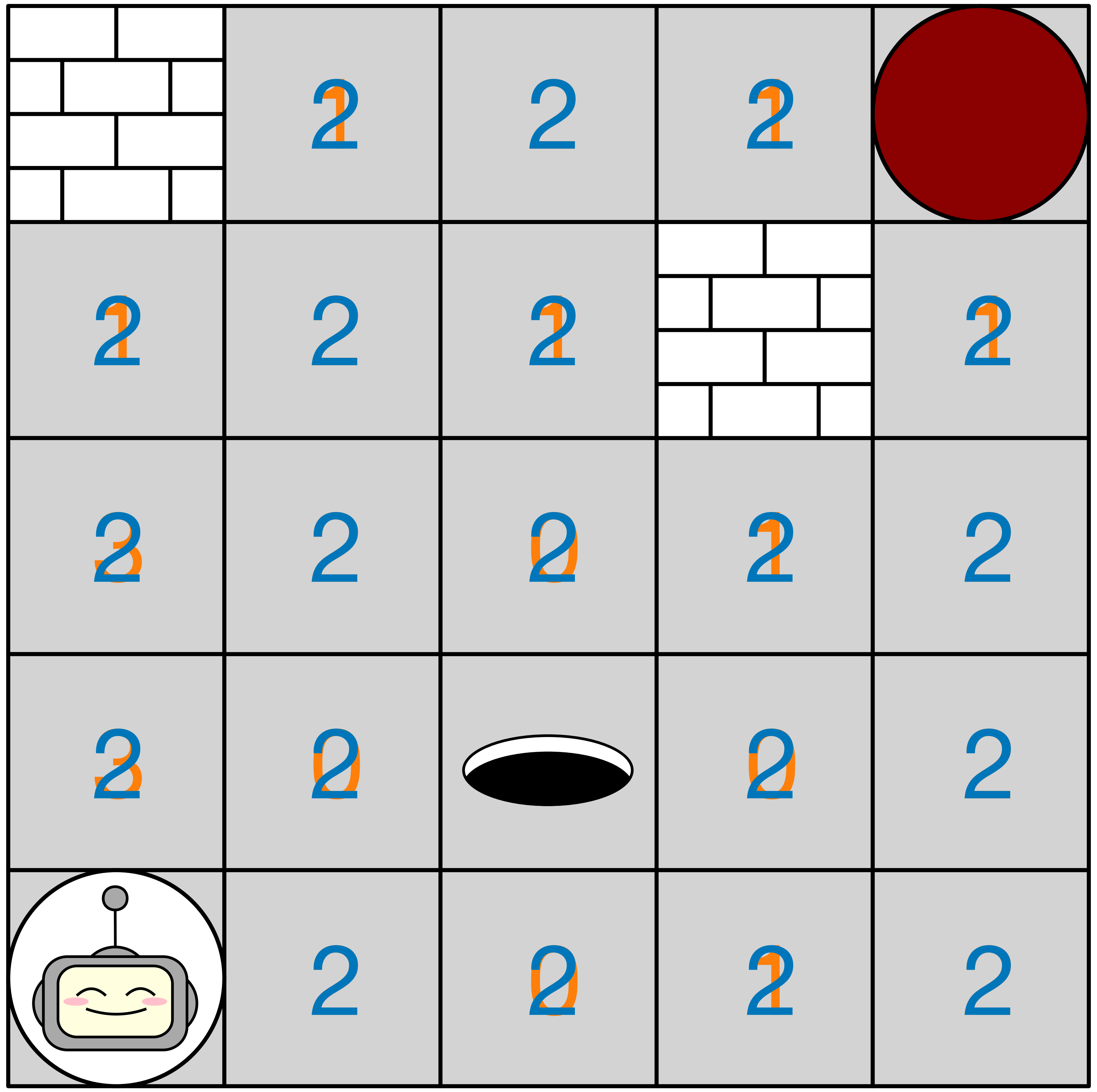
  - Adapt frequency!

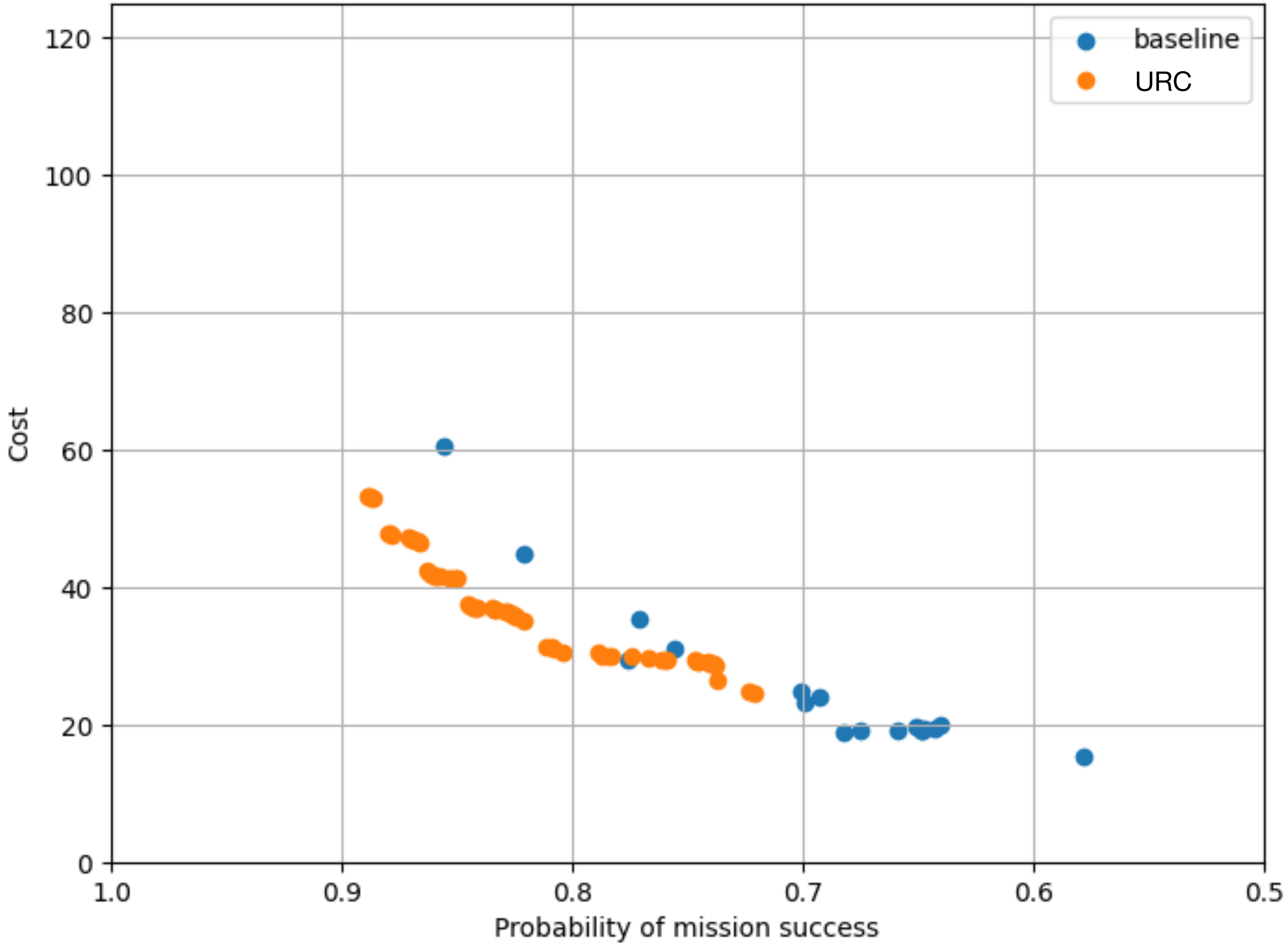


How do we derive a URC?

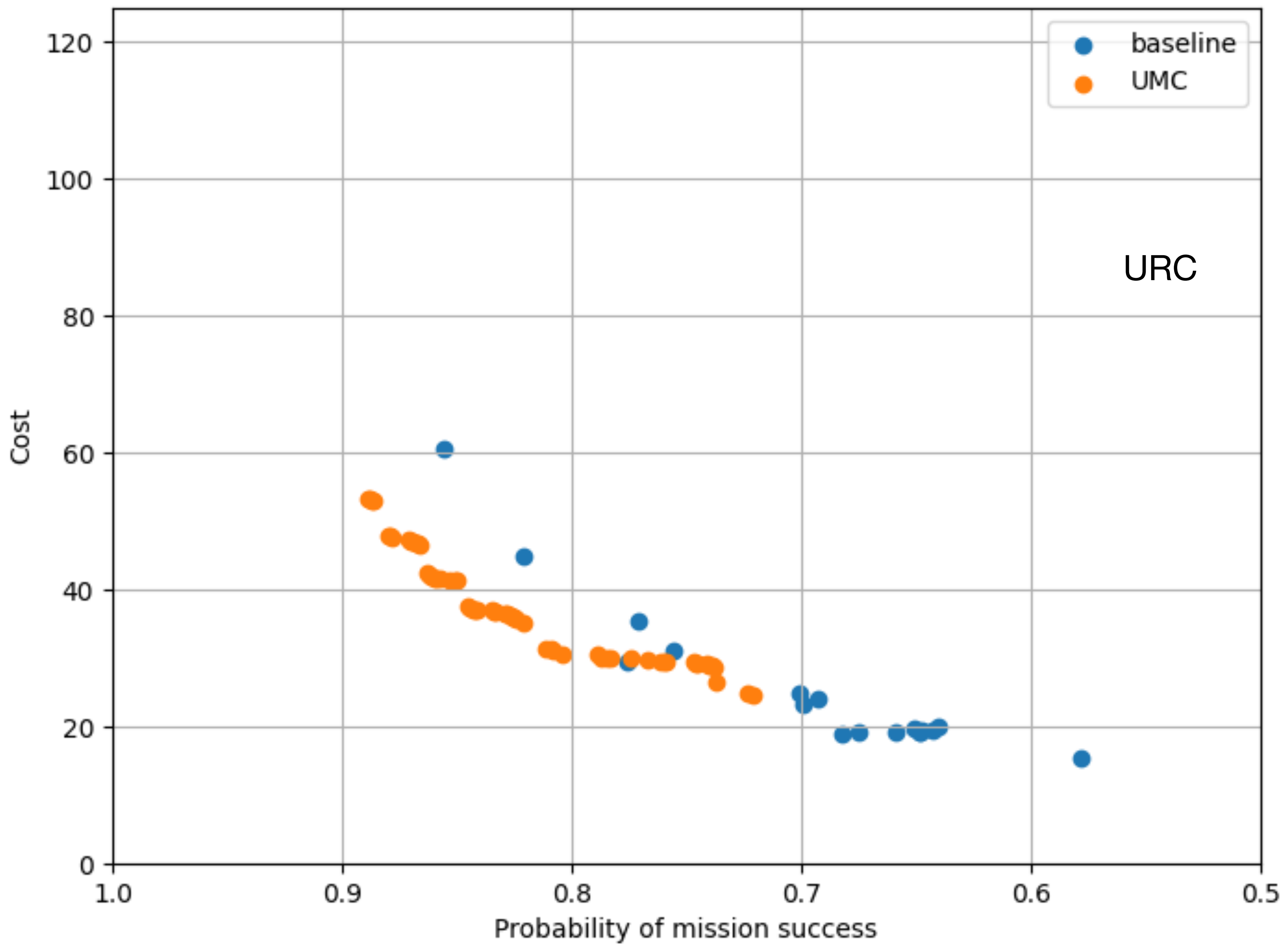
- Synthesis at design time
- Assumptions:
  - System with services and belief
  - Formal model (Markov chain) with ground truth and objectives
- Explorative search of solution space with EvoChecker

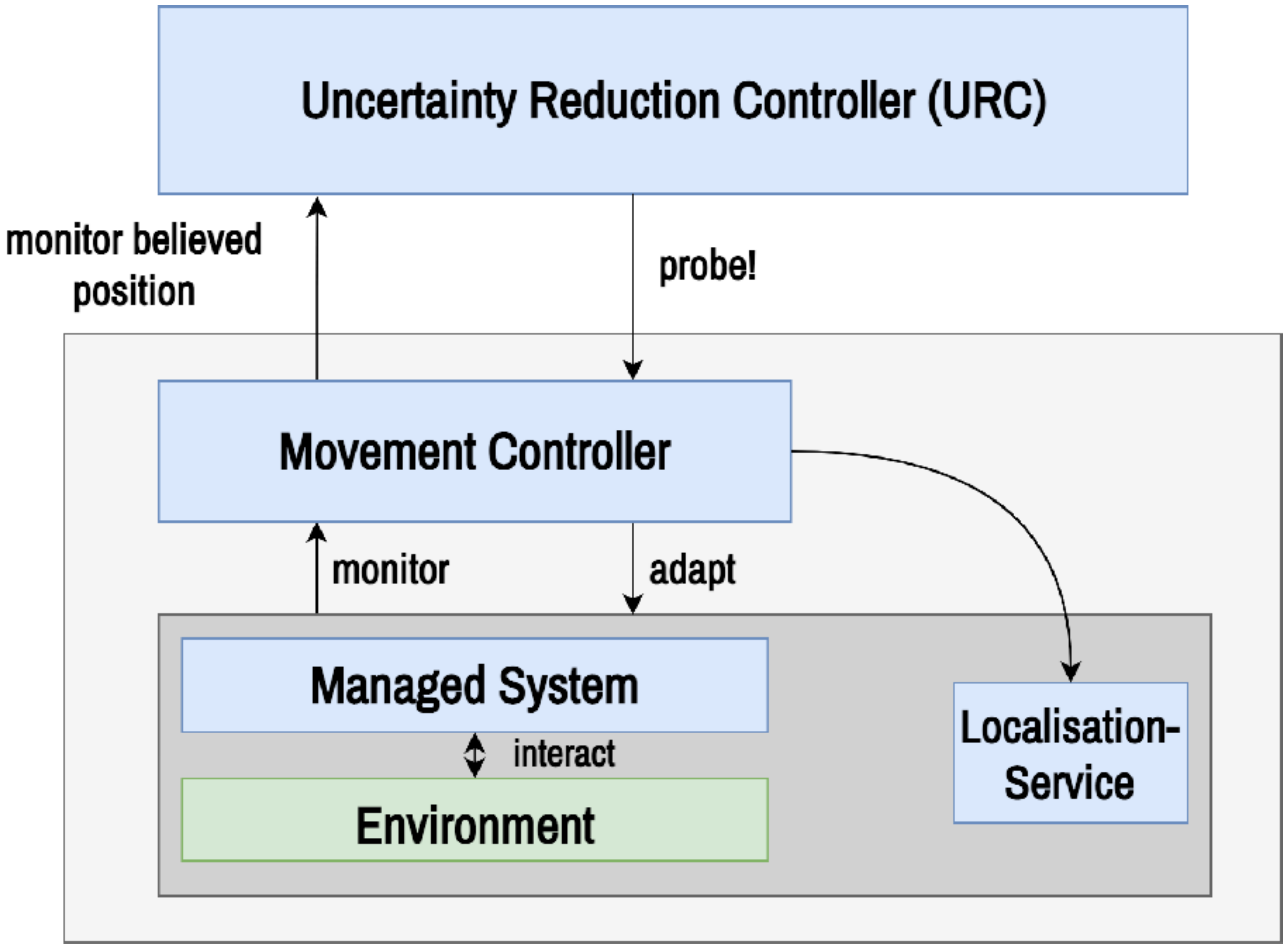
Pareto-front of solutions





Formal Synthesis of Uncertainty Reduction Controllers





carwehl@informatik.hu-berlin.de