



**Adaptation
&
Boundaries**

**Bashar
Nuseibeh**

And to conclude ...

Part 1

- The notion of a **boundary** is prevalent in software and systems engineering.
- It is sometimes a constraint, sometimes a feature, but almost **always present**.
- Perhaps considering it more explicitly in our work can be **instructive, even useful**.

Part 2

- Engineering security & privacy is often about identifying/**managing boundaries**
- This is challenging for mobile & ubiquitous systems: boundaries **are unclear & changing**.
- Engineering adaptive security & privacy can help, but these systems must be **adaptive by design** and **adaptive at runtime**.

Discipline Boundaries: Software Engineering & Engineering Design



- Technology transfer across discipline boundaries

- Multiple Perspectives (Viewpoints)



- But Overlapping boundaries



Requirements & Systems Engineering

A common misconception is that requirements engineering (RE) is all about eliciting and specifying stakeholder wishes, needs & goals

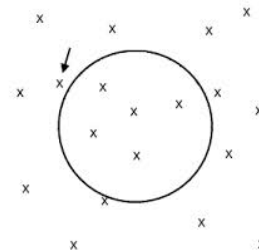
But actually, as Jackson has argued, RE is first and foremost about **problem framing** (identification, formulation & bounding).

Requirements



■ A key to this is eliciting **problem boundaries (scope)**:

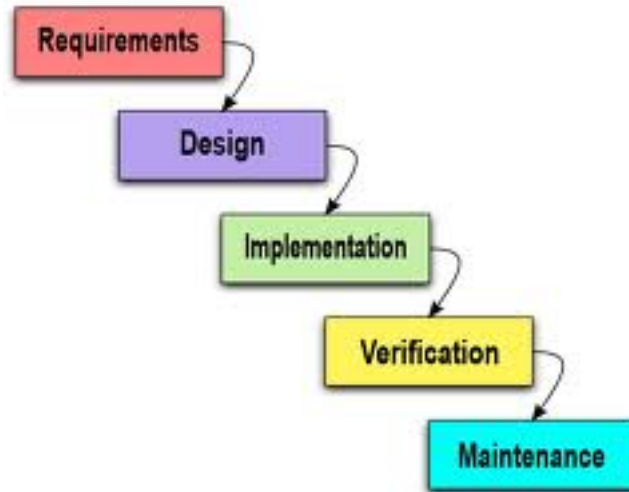
- Too wide → too much to do
- Too narrow → lost opportunity



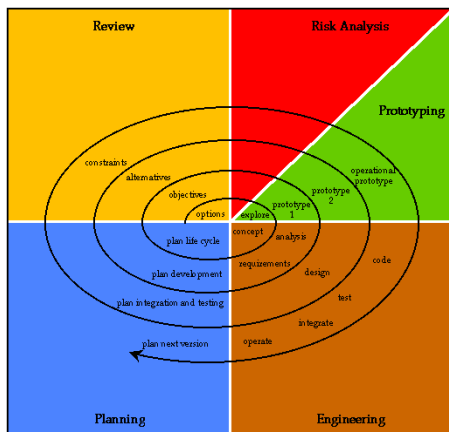
■ **Boundary critique** (Ulrich 2002, following Churchman 1970)

Boundaries & the Software Development Process

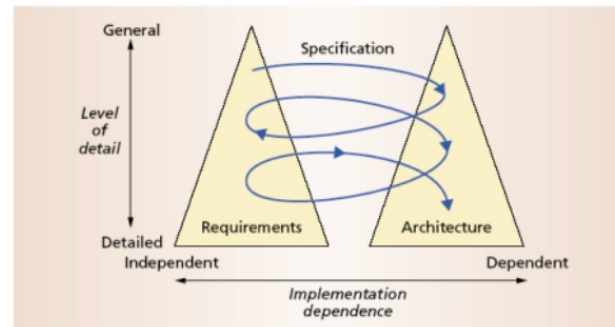
■ Waterfall



■ Spiral and Twin Peaks

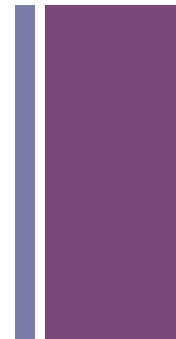


Twin peaks model



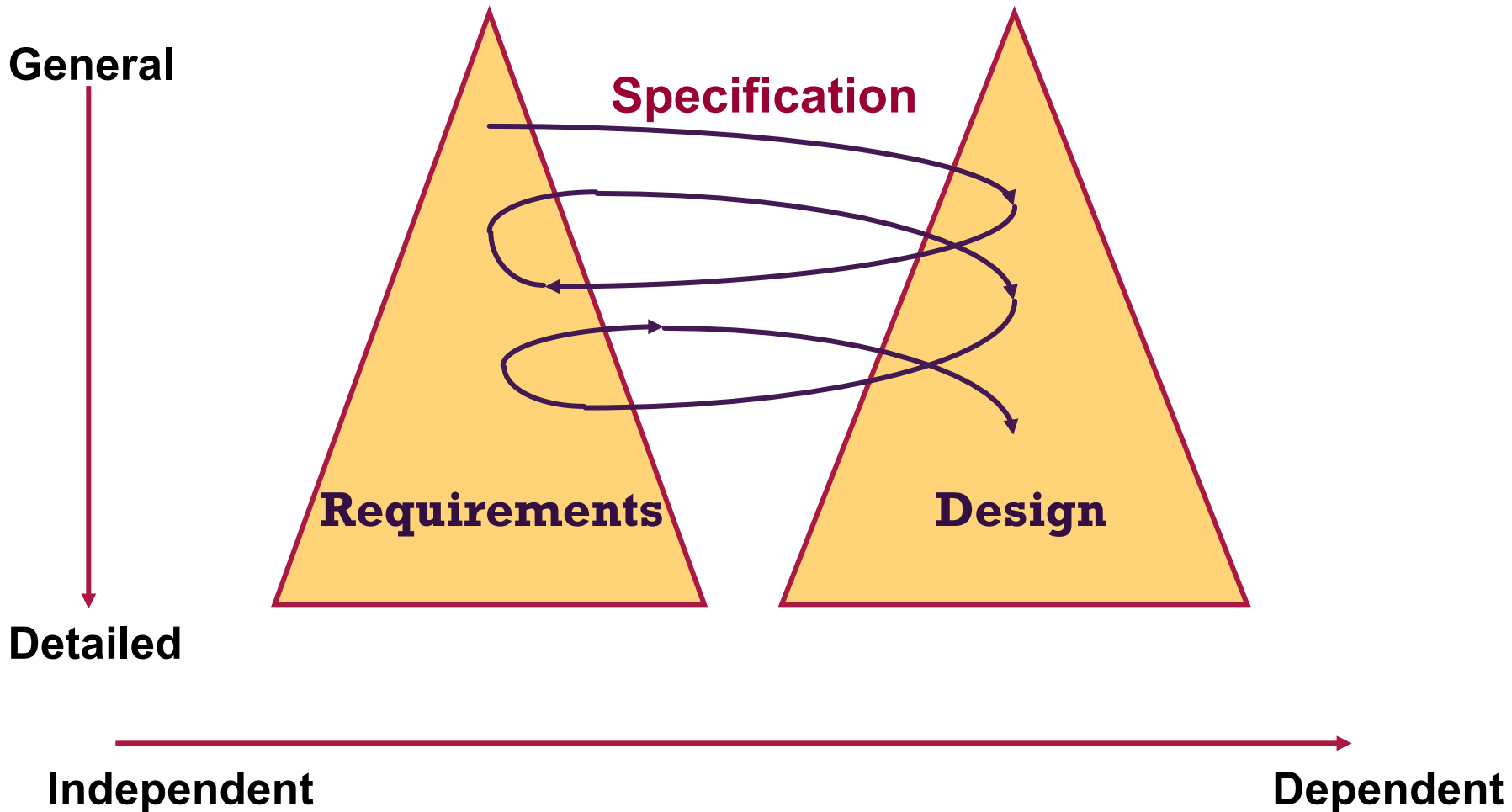


Disappearing Boundaries?



- **Some argue that many boundaries are ‘disappearing’, eg:**
 - *Adaptive systems*: The disappearing boundary between development-time and run-time [Baresi & Ghezzi 2010]
 - *Software Process*: Weaving requirements & architecture [Nuseibeh 2001]
 - *End user programming* [Burnett and others 2003]
 - *Interaction Design*: Making vs Using [Nakakoji 2011]
- **But actually the boundaries are still there**
 - Tacit or explicit
 - And the challenge is to identify them and better understand and manage what it is that they separate.

Twin Peaks



Security and Boundaries



Choosing the wrong boundary ..

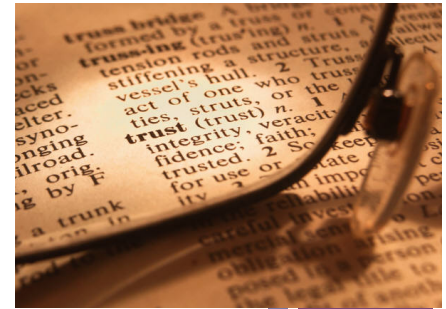


In a hotel room..

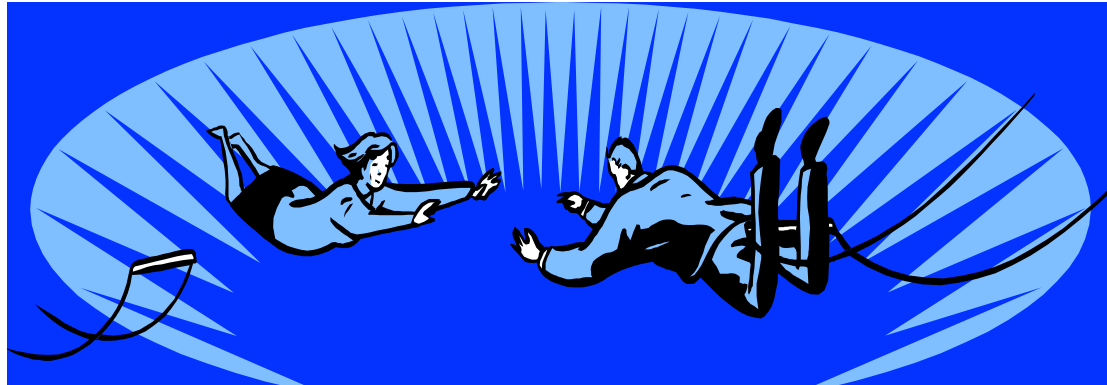


This is a demo only!

Trust Assumptions



- Are the raw materials of the problem boundary



- Are assumptions affecting security, believed to be true
 - There may be no solid evidence, perhaps even no thought.
 - Assumption may be about behavior
 - Assumption may be the existence or non-existence of a domain

Trust Assumptions and Boundaries

“Only authorized cars on site”

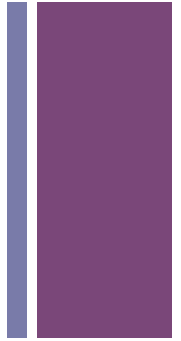


“No valuable assets to protect”

“Drivers will behave responsibly”

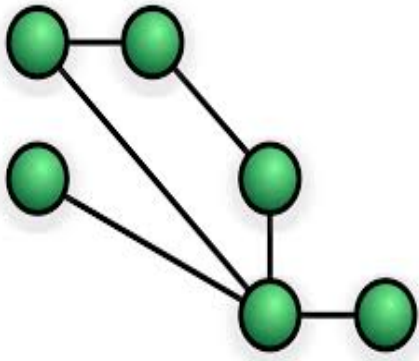


Changing Boundaries

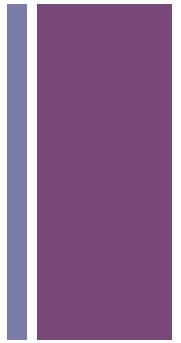


- **Mobile and Ubiquitous Computing** requires “seamless” integration of people and technologies (eg, Smart Homes, Cities & the IoT)
- **Adaptive systems** promise to deliver such integration.
- **But the boundaries** between the mobile devices, the infrastructure, and people are becoming increasingly **difficult to identify and manage** as people and devices move.
- Trust Assumptions that used to bound security problems no longer hold.
- And here lies the challenge of **adaptive security...**



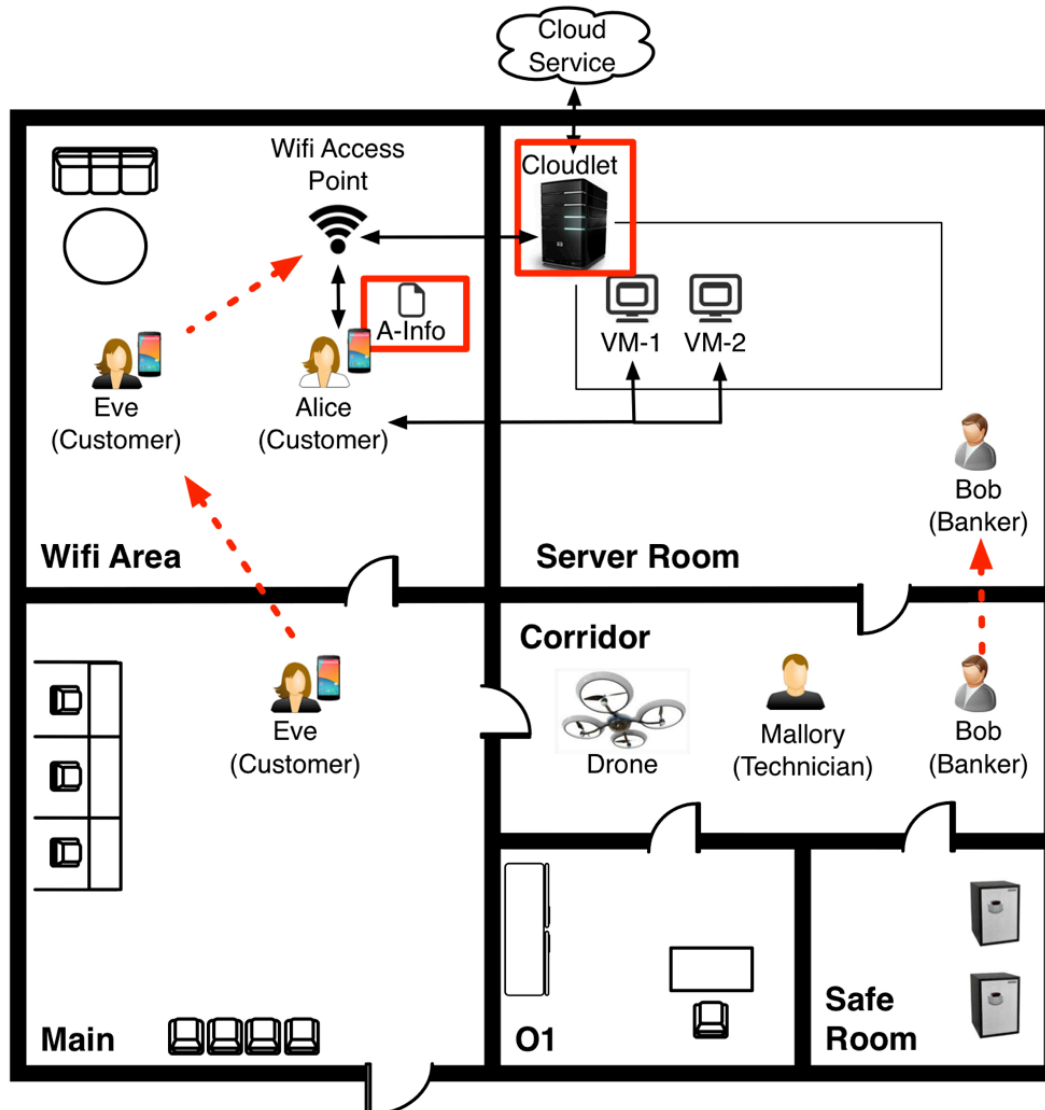


Topological Boundaries



- **Topology** denotes the **structure of the operational environment of a system**, including assets, agents, and their relationships.
- Topology provides a **richer representation of context** (and its boundaries) that can help engineer adaptive security
 - And may also help engineer adaptive systems more generally.
- Topology represent be **physical, digital and/or social** spaces
 - And a key challenge is to understand and manage the interplay between them (across their boundaries)

Topology Aware Adaptive Security



Physical and cyber spaces of a modern bank branch

References:

Pasquale et al @ SEAMS'12

Tsigkanos et al @ RE'14

Tsigkanos et al @ [ICSE'15](#)

Thanks and Acknowledgements



Funding thanks to: ERC, EPSRC, The Royal Society, SFI, QNRF, NII