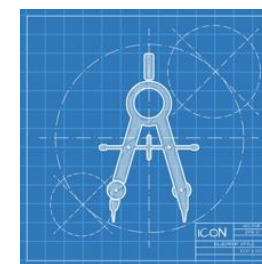
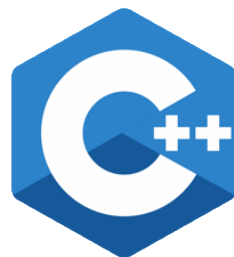


Tilting at White Towers

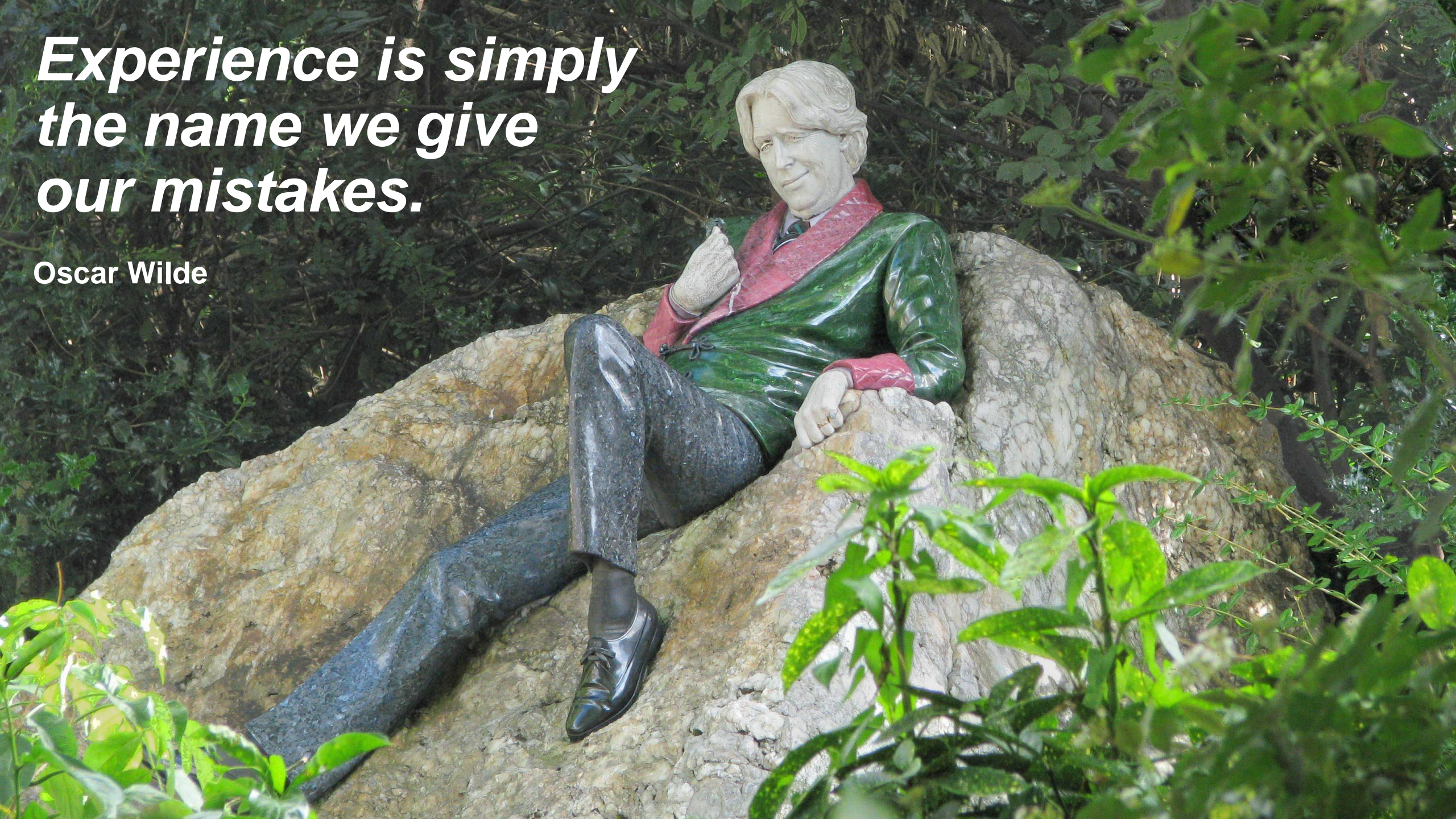
Making your identity architecture actionable

A little bit about Hutch

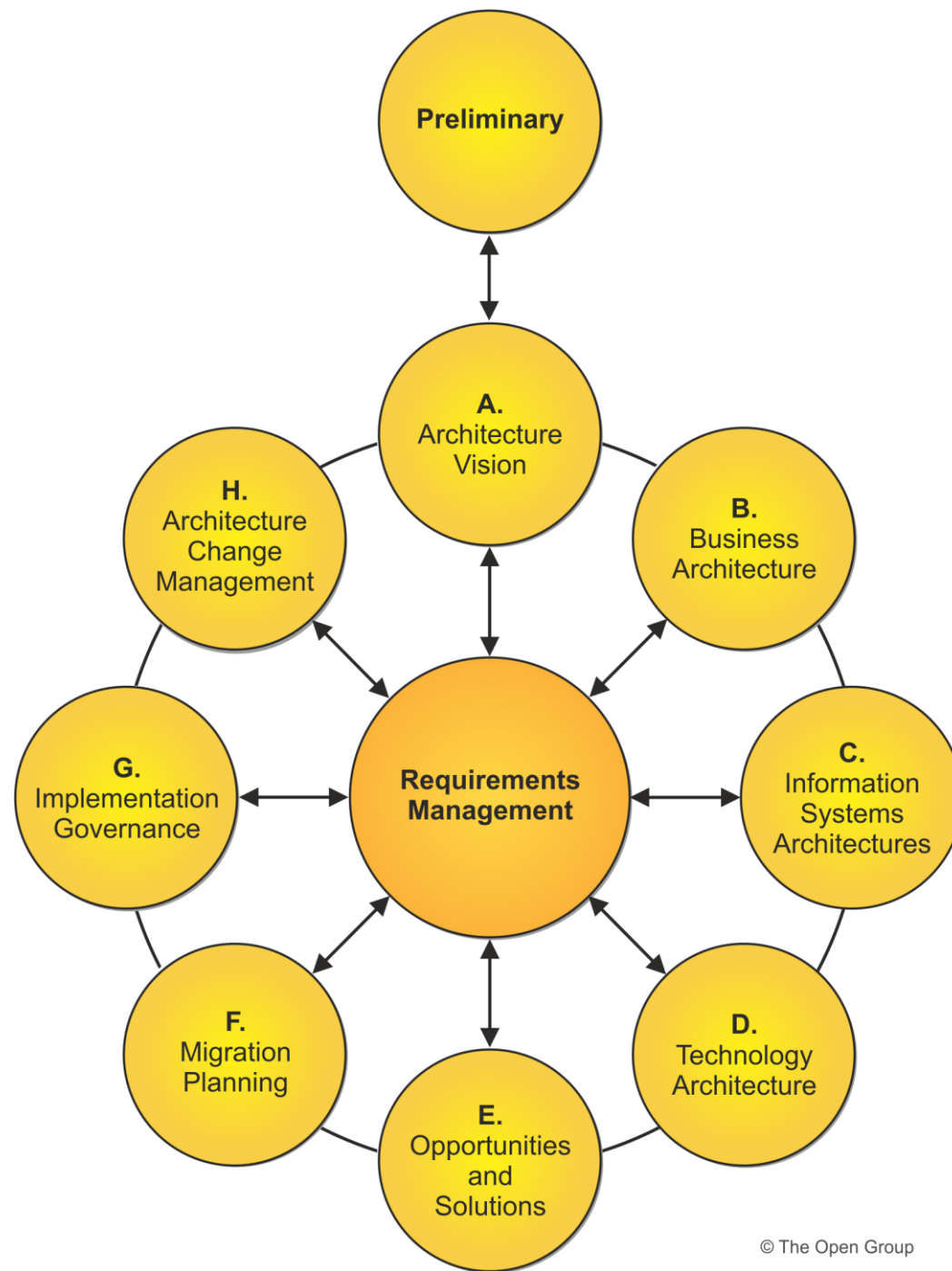


***Experience is simply
the name we give
our mistakes.***

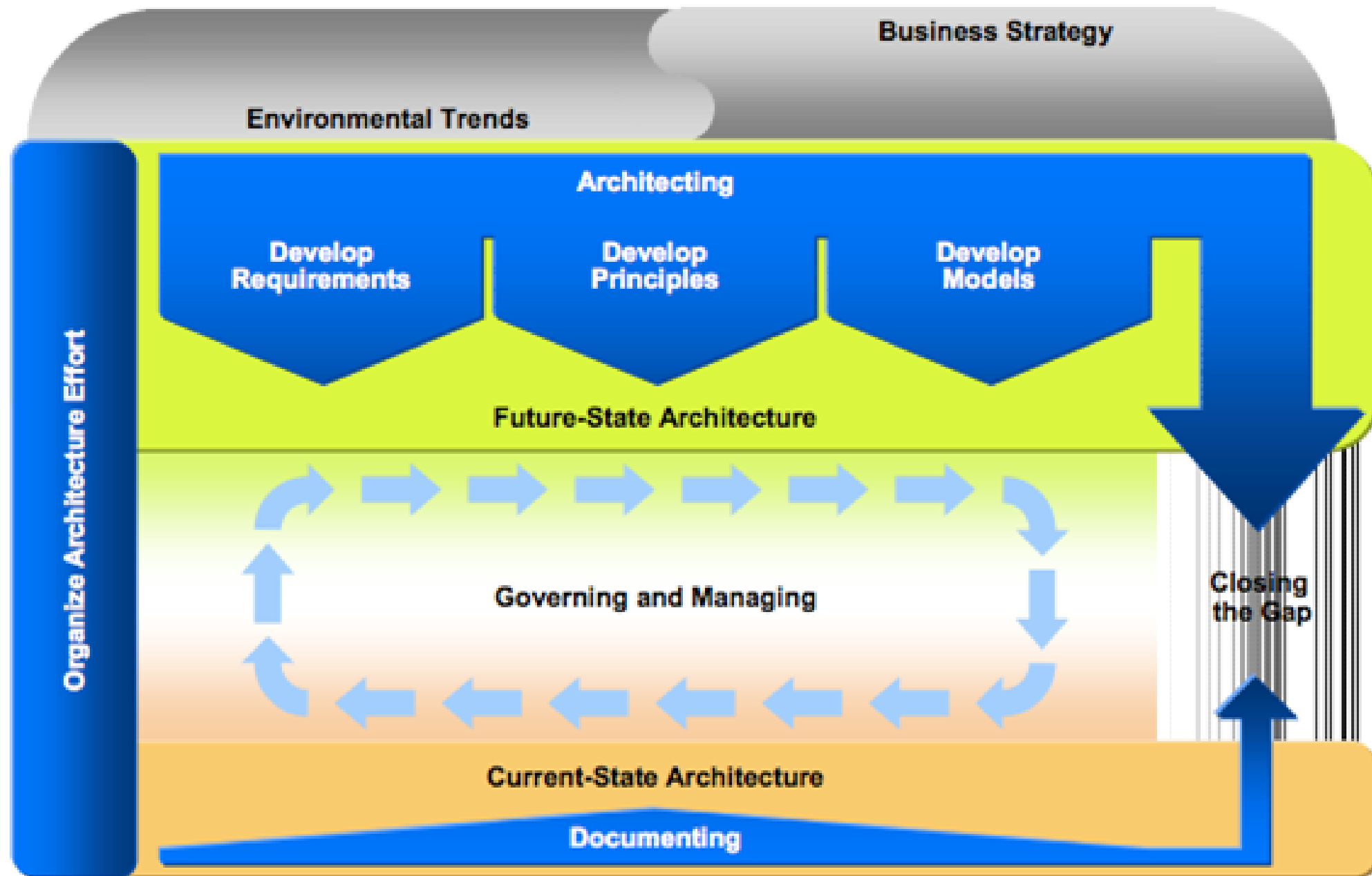
Oscar Wilde

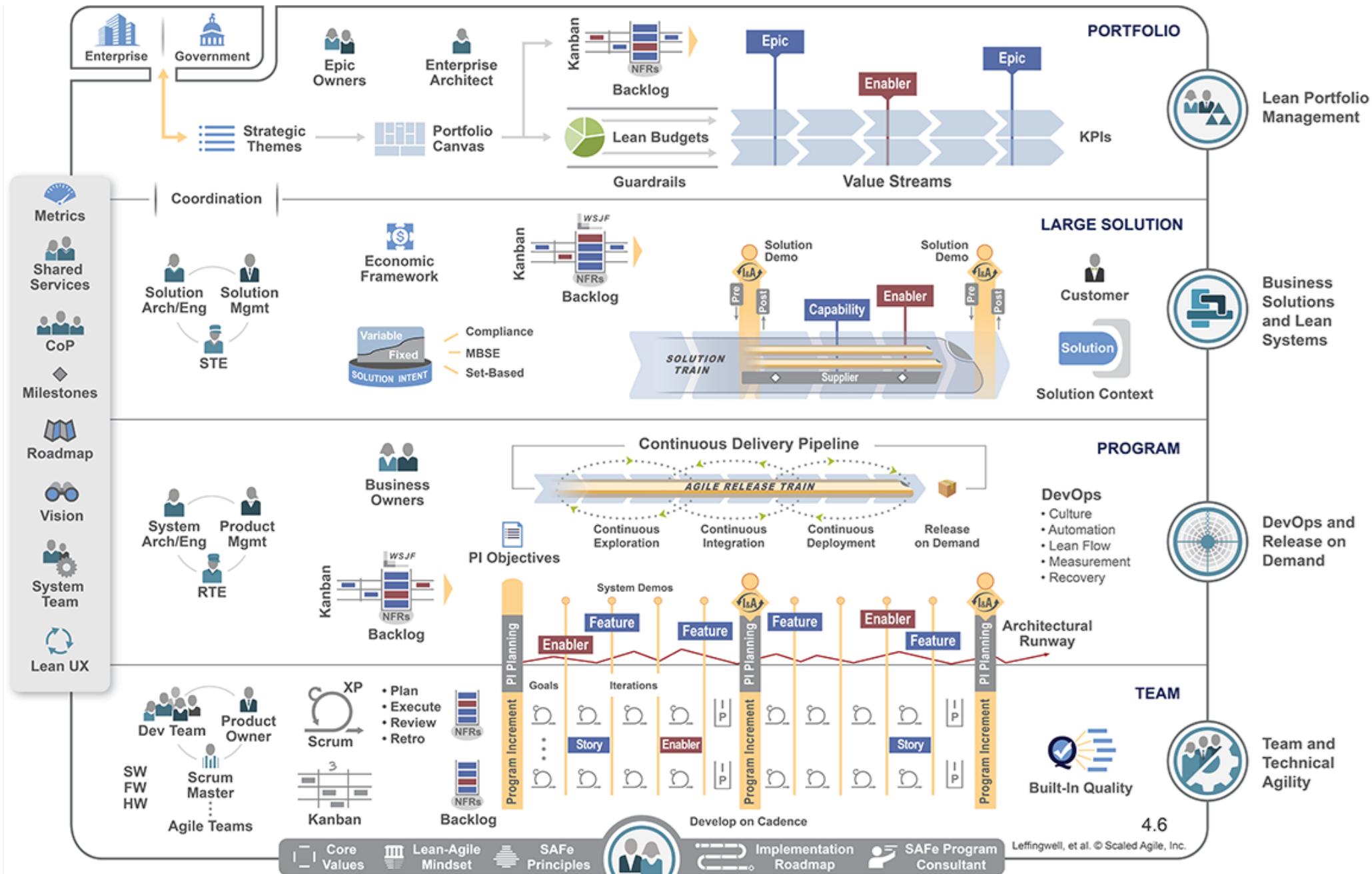


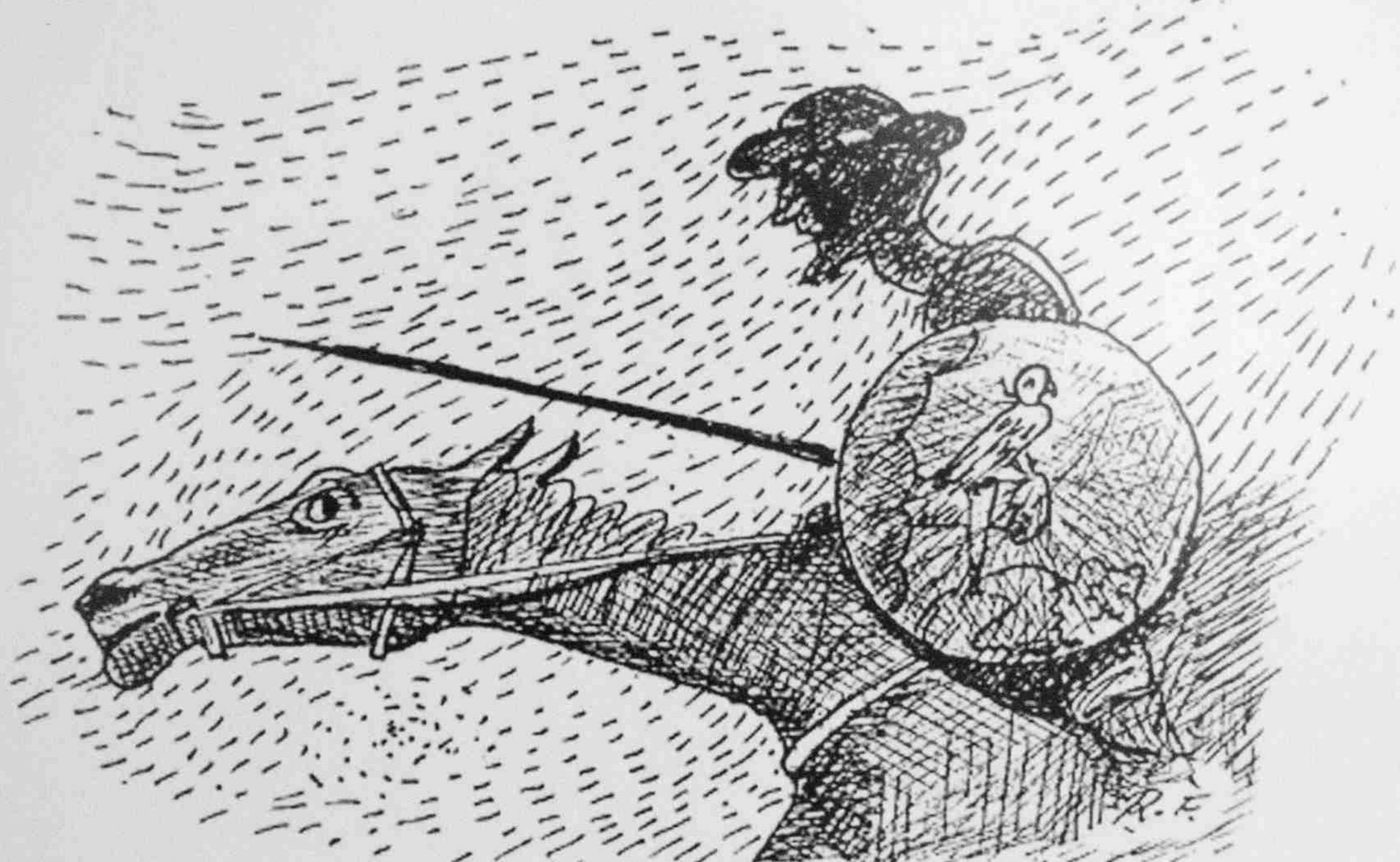
	DATA <i>What</i>	FUNCTION <i>How</i>	NETWORK <i>Where</i>	PEOPLE <i>Who</i>	TIME <i>When</i>	MOTIVATION <i>Why</i>
Objective/Scope (contextual) <i>Role: Planner</i>	List of things important in the business	List of Business Processes	List of Business Locations	List of important Organizations	List of Events	List of Business Goal & Strategies
Enterprise Model (conceptual) <i>Role: Owner</i>	Conceptual Data/ Object Model	Business Process Model	Business Logistics System	Work Flow Model	Master Schedule	Business Plan
System Model (logical) <i>Role: Designer</i>	Logical Data Model	System Architecture Model	Distributed Systems Architecture	Human Interface Architecture	Processing Structure	Business Rule Model
Technology Model (physical) <i>Role: Builder</i>	Physical Data/Class Model	Technology Design Model	Technology Architecture	Presentation Architecture	Control Structure	Rule Design
Detailed Representation (out of context) <i>Role: Programmer</i>	Data Definition	Program	Network Architecture	Security Architecture	Timing Definition	Rule Speculation
Functioning Enterprise <i>Role: User</i>	Usable Data	Working Function	Usable Network	Functioning Organization	Implemented Schedule	Working Strategy

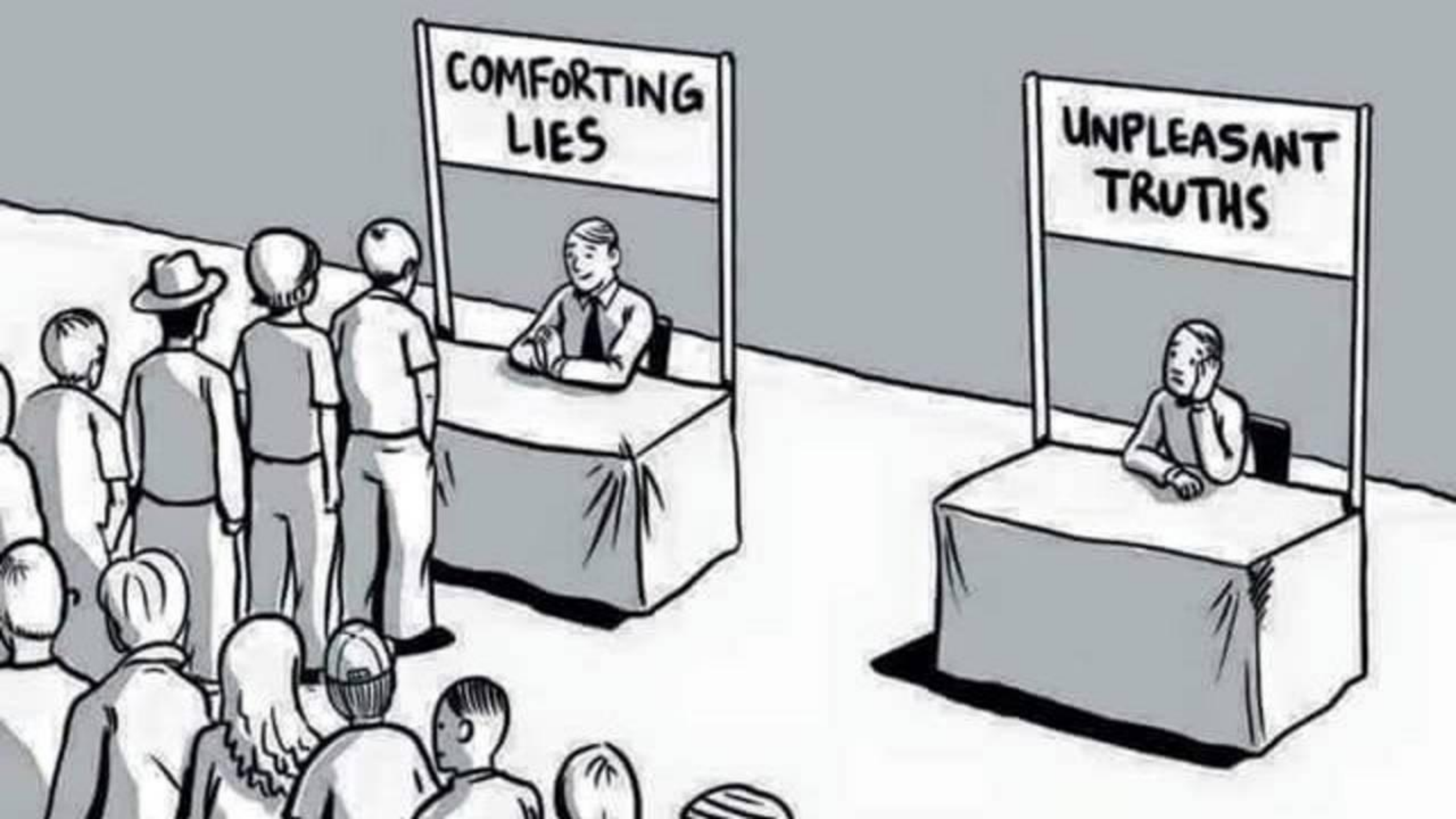


	DATA <i>What</i>	FUNCTION <i>How</i>	NETWORK <i>Where</i>	PEOPLE <i>Who</i>	TIME <i>When</i>	MOTIVATION <i>Why</i>
Objective/Scope (contextual) <i>Role: Planner</i>	List of things important in the business	List of Business Processes	List of Business Locations	List of important Organizations	List of Events	List of Business Goal & Strategies
Enterprise Model (conceptual) <i>Role: Owner</i>	Conceptual Data/ Object Model	Business Process Model	Business Logistics System	Work Flow Model	Master Schedule	Business Plan
System Model (logical) <i>Role: Designer</i>	Logical Data Model	System Architecture Model	Distributed Systems Architecture	Human Interface Architecture	Processing Structure	Business Rule Model
Technology Model (physical) <i>Role: Builder</i>	Physical Data/Class Model	Technology Design Model	Technology Architecture	Presentation Architecture	Control Structure	Rule Design
Detailed Representation (out of context) <i>Role: Programmer</i>	Data Definition	Program	Network Architecture	Security Architecture	Timing Definition	Rule Speculation
Functioning Enterprise <i>Role: User</i>	Usable Data	Working Function	Usable Network	Functioning Organization	Implemented Schedule	Working Strategy









COMFORTING
LIES

UNPLEASANT
TRUTHS

Uncomfortable Truth #1:

Architecture focuses too much on the future state





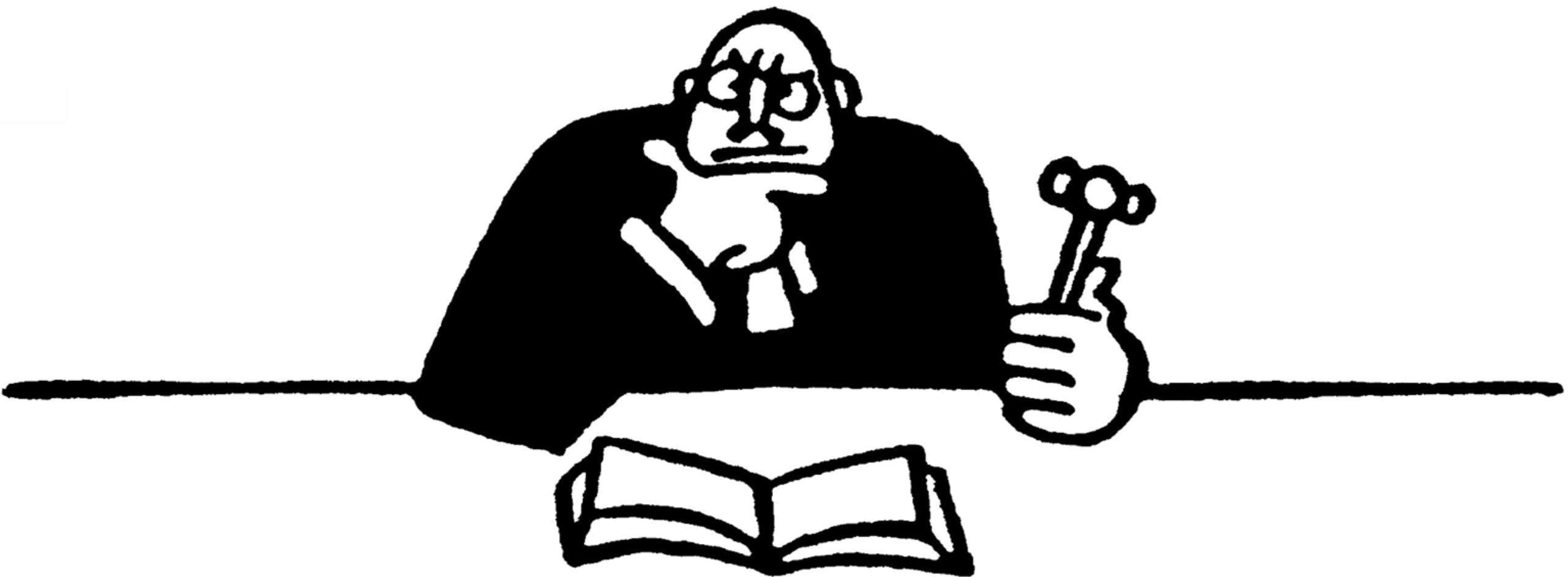
So why does this happen?

- The technology infrastructure that we have at some point doesn't really fit the technology infrastructure that we want or need
- Business goals are far more fluid than the architecture can be
- The future state does not define our actions but actions taking place by the business will define our future state

Uncomfortable Truth #2:

**Architecture principles honestly have
little effect**

and can actually hurt your efforts

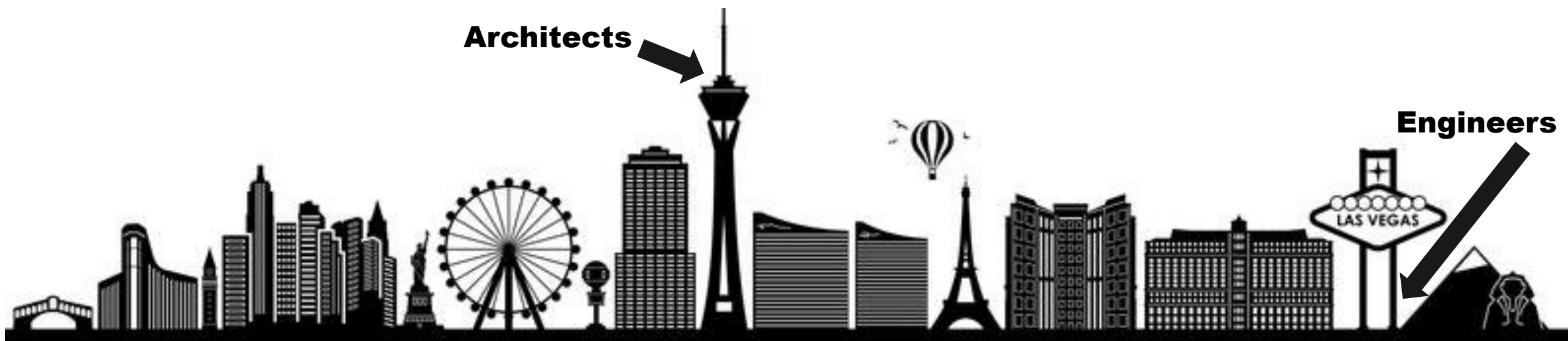


So why does this happen?

- Architecture principles have been espoused as a way to make future design choices “good”
- Principles are usually written as ‘desired outcomes’ but that doesn’t mean following them gets you that outcome
- The idea that a set of design rules can get you a well-defined landscape is simply too good to be true

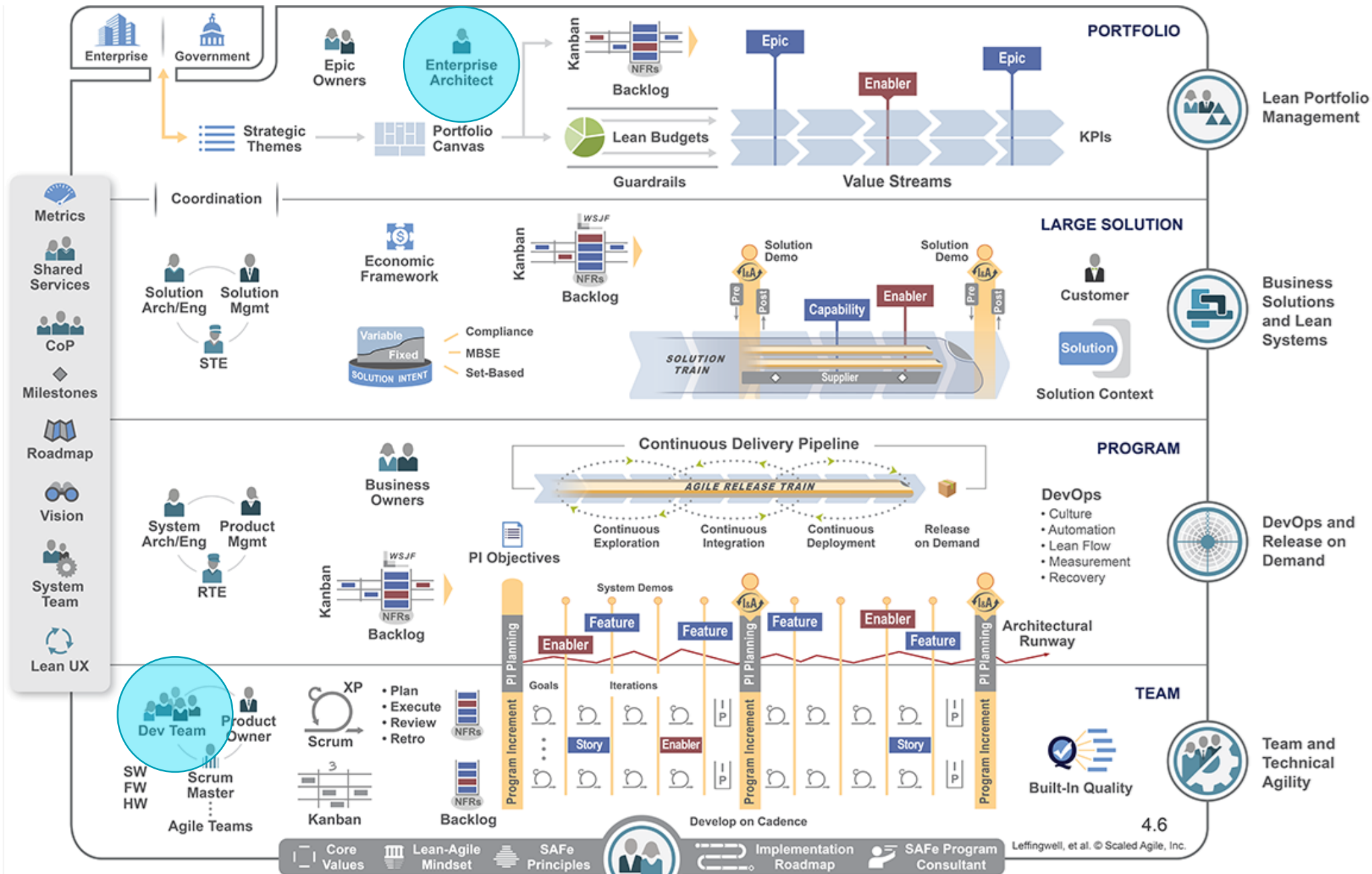
Uncomfortable Truth #3:

Architects are often far removed from front-line engineering teams



Architects

Engineers

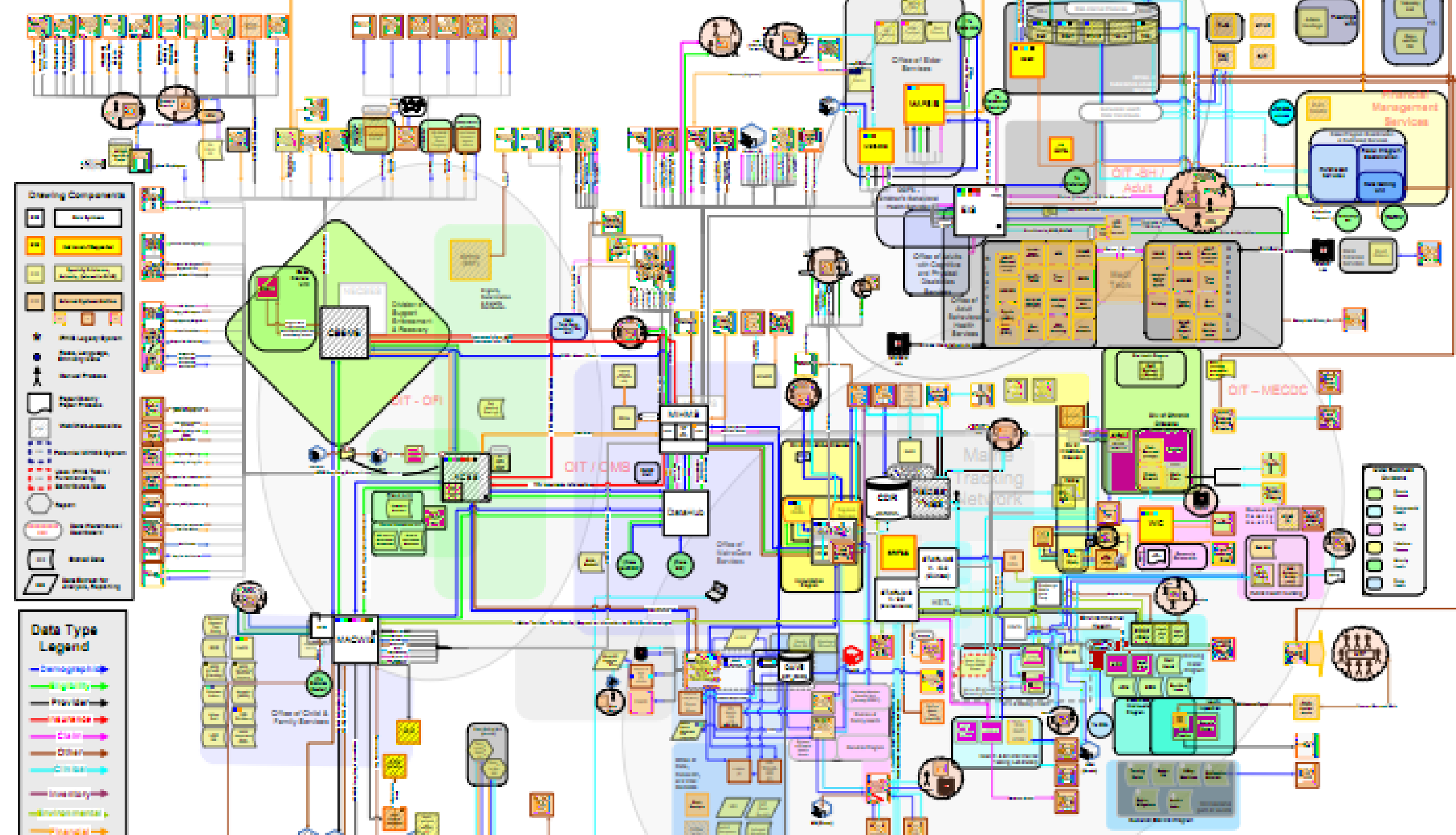


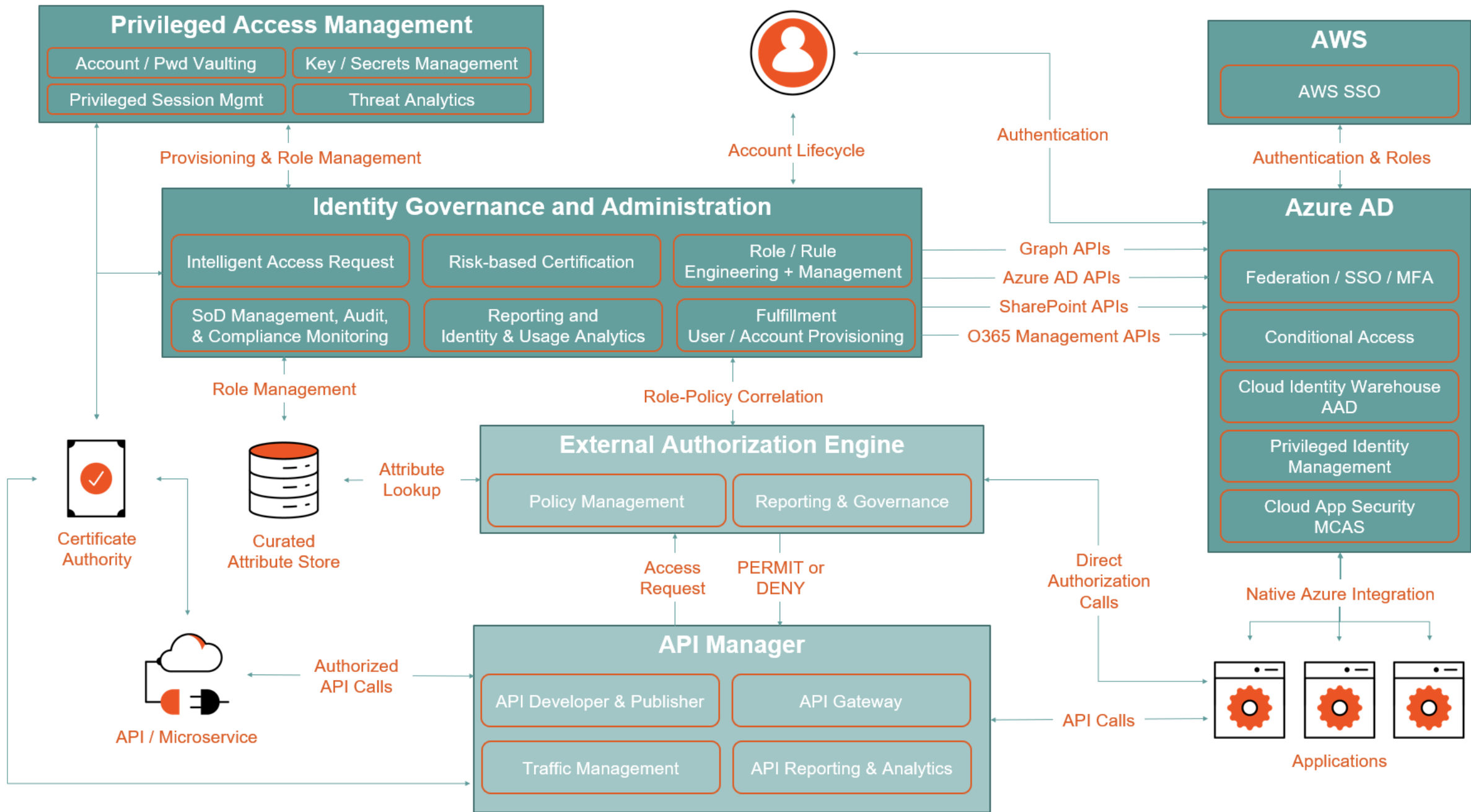
So why does this happen?

- Organizations do not stumble over mountains, they stumble over molehills
- Frameworks do not create architectures, people do
- Good architects have deep domain knowledge that allow them to see all the possible good solutions, and to also recognize the bad ones.

Uncomfortable Truth #4:

Most architecture diagrams are not very useful





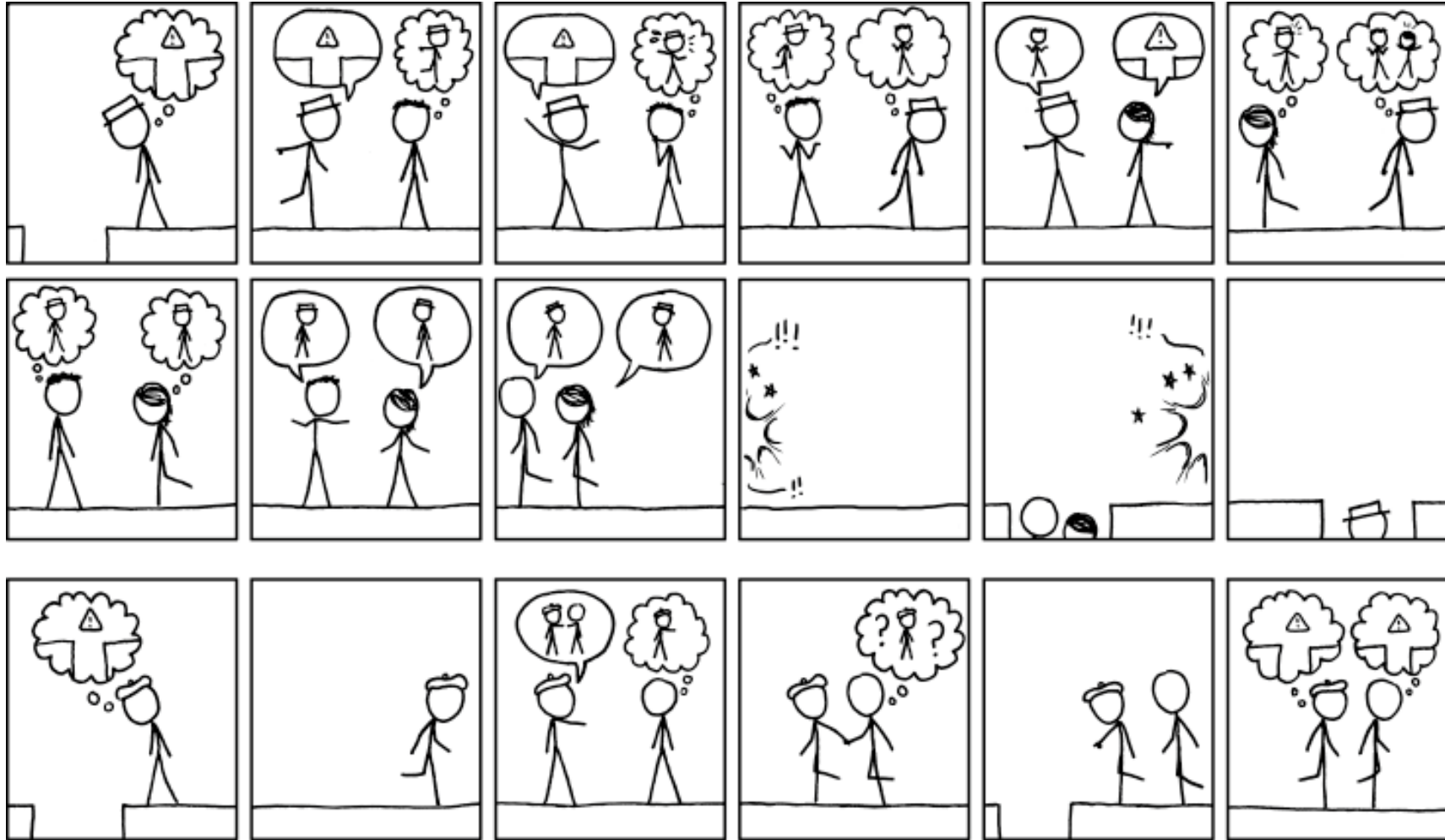
So why does this happen?

- Architects need to target the audience, not a methodology
- Less is more
- There should be a relationship between the abstractions and conventions laid down by architecture and those actually used in the system

Uncomfortable Truth #5:

Architecture needs to communicate more openly with its stakeholders

Anyone who says they're great at communicating but 'people are bad at listening'



is confused about how communication works

So why does this happen?

- Governance is good, but collaboration is better
- Crowdsourcing knowledge ensures that important information is captured from previously unrecognized sources
- Architects should be aligned to a small set of engineering teams, who they empower to make architectural decisions

So ... now what?



I'M DROWNING

AND YOU'RE DESCRIBING THE WATER

Here are some life rings

- Make architecture more agile
- Open up the conversation with technical engineering teams
- Transition your architecture effort from a 'project' to a 'product'
- Bring decentralization & democratization to your architecture



THANK YOU!



Steve “Hutch” Hutchinson

Director of Security Architecture

Mitsubishi Bank of Tokyo