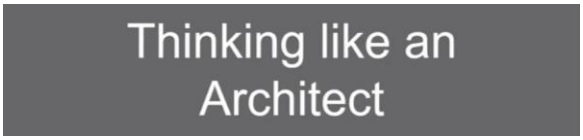
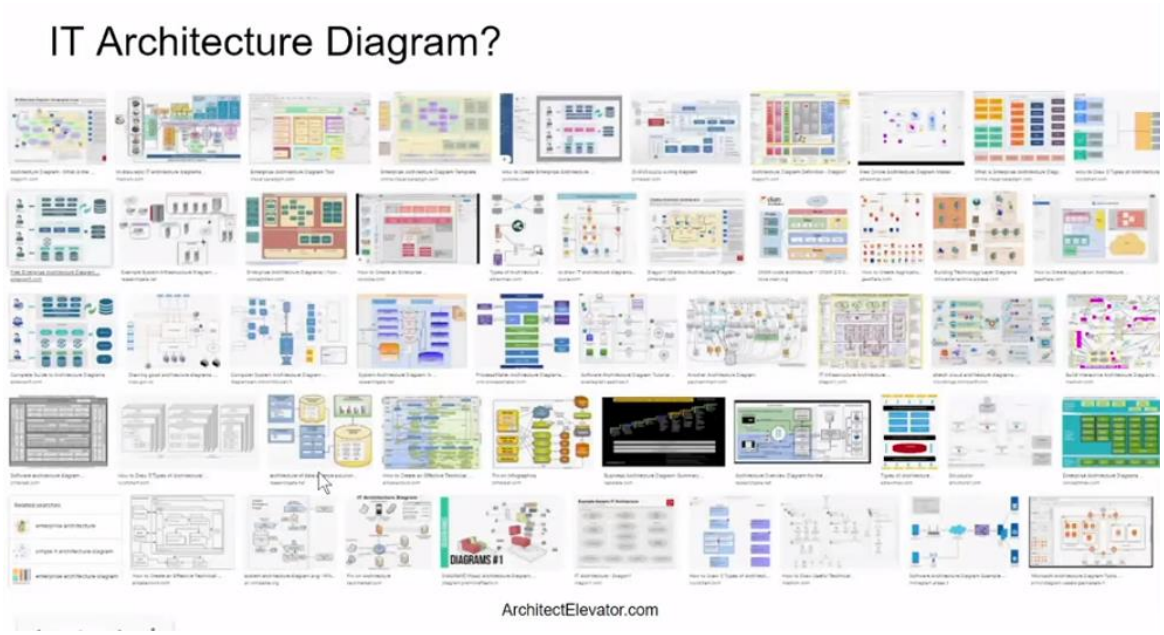
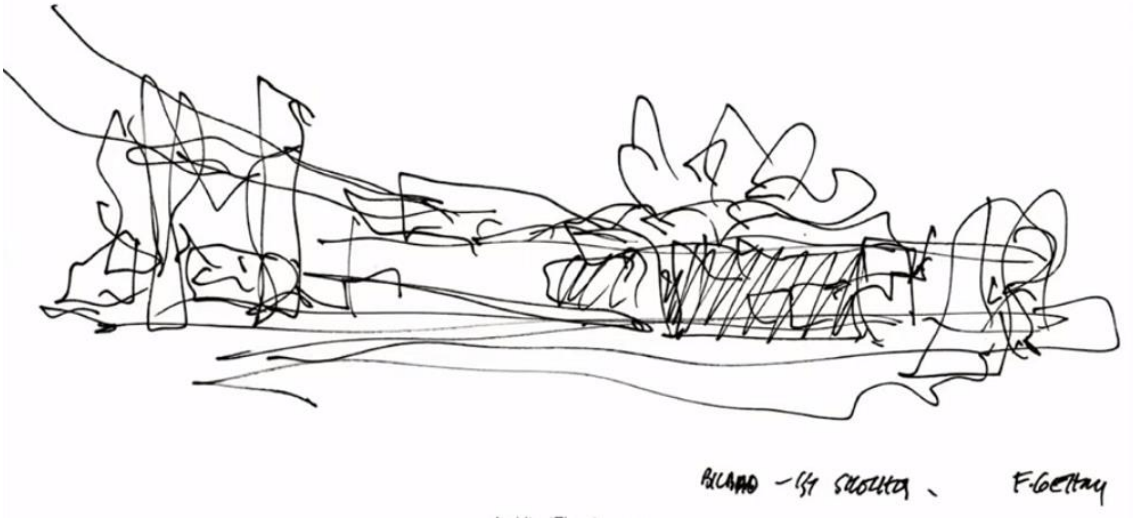
Platforms have fueled some of the most successful business models of the past decade. However, building one for your enterprise isn't as simple as it looks - many platform initiatives are doomed from the start. This keynote from YOW! September 2021 looks behind the buzzword to identify what makes platforms work and how you can successfully build an in-house platform.



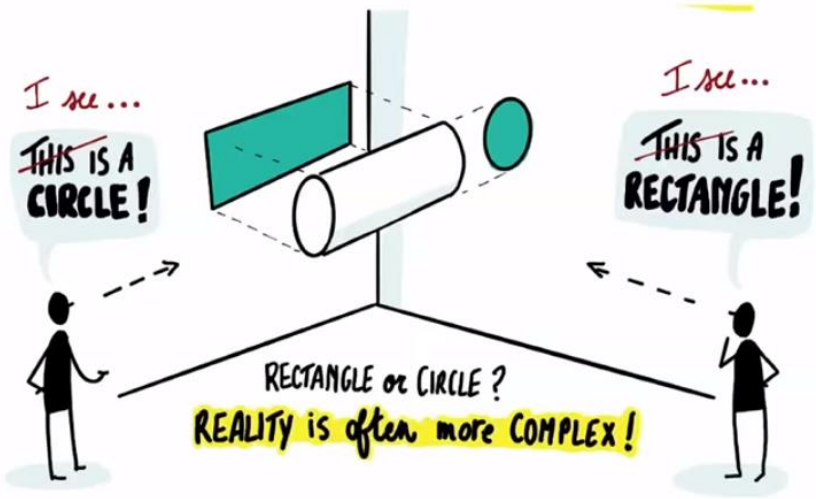


Famous Architects Sketch



Famous Architects Sketch

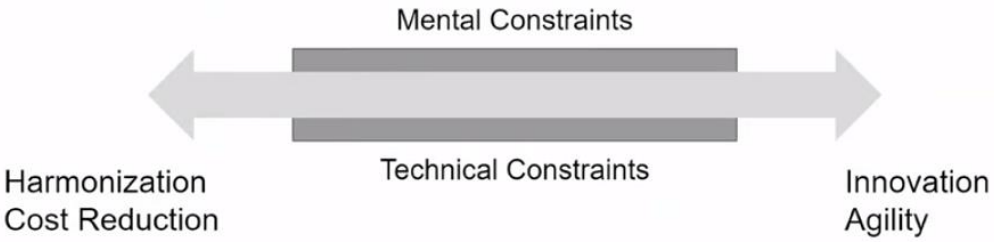
Architects See More Dimensions



Architects Zoom in and Out, See Different Things

Cloud Architecture

Cloud = Removing Constraints

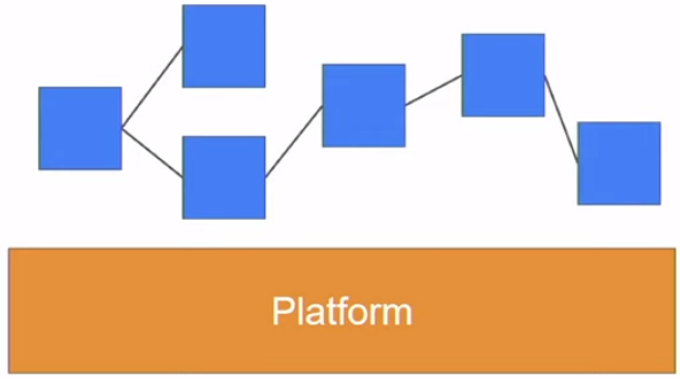


Opposites Attract

Quality	⟷	Speed
Standardization	⟷	Innovation
Customization	⟷	Cost
Automation	⟷	Control
Openness	⟷	Monetization

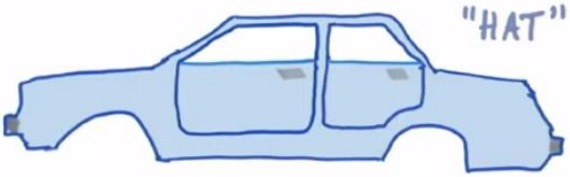
Platforms - Giant's Shoulders

Platform = Broadly Used Components



Automotive Platforms

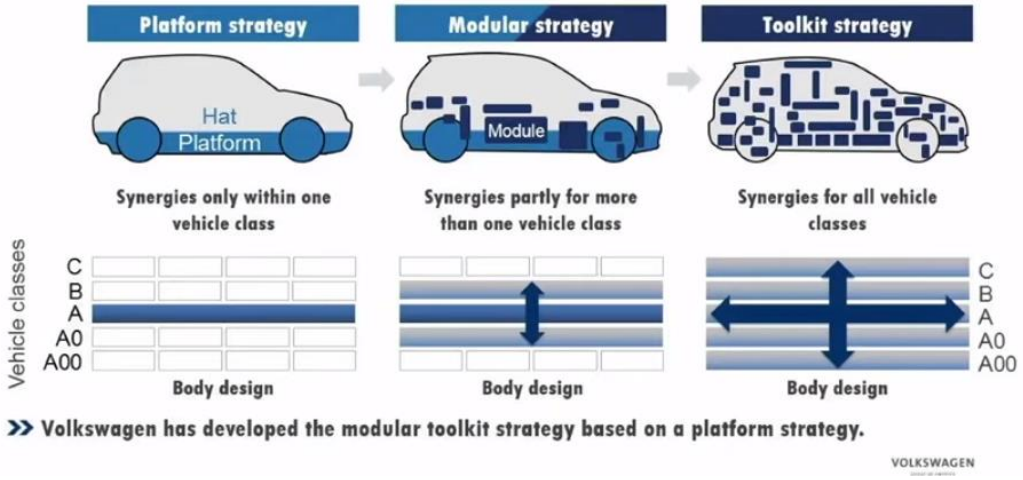
- Lower investment
- Diversified
- “Differentiated”



- High investment
- Standardized
- “Undifferentiated”

Successful Automotive Platforms - Volkswagen MLB Evo

From platform to toolkit (“Baukasten”)



The Platform Paradox

Standards Enable

“If men are to accomplish together anything useful whatever they must, above all, be able to understand one another. That is the basic reason for a National Bureau of Standards.”

MEASURES FOR PROGRESS - A HISTORY OF THE NATIONAL BUREAU OF STANDARDS
US Dept. of Cpmmerce 1966

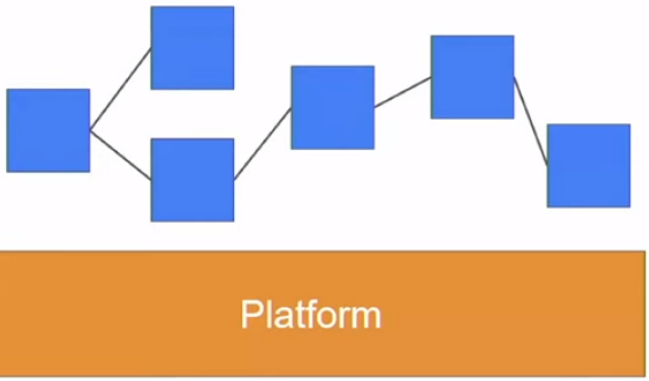
"A4 Paper doesn't stifle creativity"

"Platforms are a means of centralizing expertise while decentralizing innovation to the customer or user."

Peter Gillard-Moss, ThoughtWorks

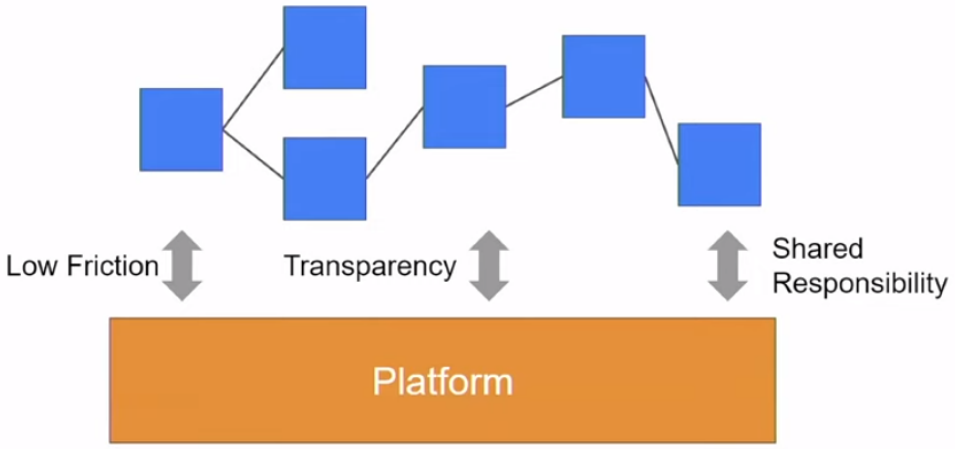
Platform Characteristics

Platform = Broadly Used Components (?)



The key point is how the different blue boxes/components interact between each other and the platform.

Platforms = More Than Just Broadly Used Components



You need low friction, transparency and shared responsibility (the platform cannot fix all the problems) when trying to use the platform.

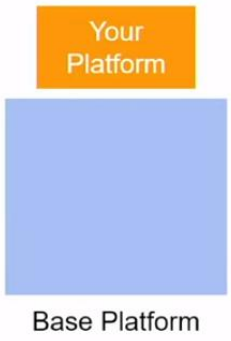
The Three "E"s of Successful Platforms

Enable	Platforms make users' lives easier, for example by speeding up software delivery
Evolve	Platforms aren't static; they evolve based on users' needs. Ideally, needs can be sensed from the platform usage
Embrace	Platforms embrace lower-level platforms instead of reinventing the wheel.

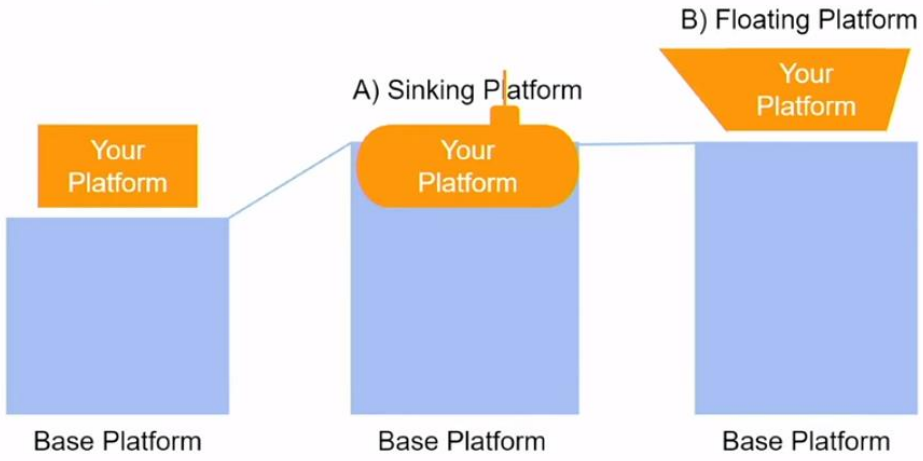
Characteristic	Platform	IT Service
Scale Effect	Thrives	Bottleneck
Marginal cost	Low	Medium to High
Friction	Low	High
Interaction	Self-Service	Ticket-based
Orientation	Customer-Centric	Process-Centric
Responsibility	Shared	Separated
Evolution	Continuous	Sporadic
Extensibility	Open / semi-open	Closed
(Adoption)	Voluntary	Mandated

Platform Architecture

Building a Platform: Floating or Sinking



Building a Platform: Floating or Sinking



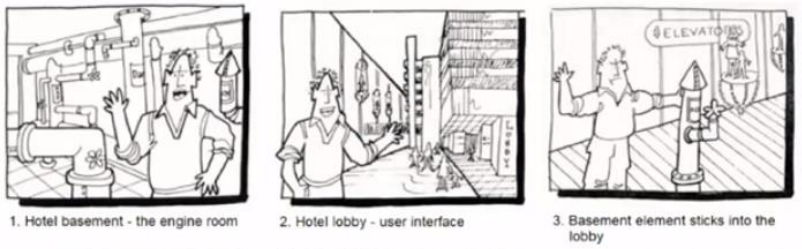
Technical Mechanisms for Platforms

Business Objective	Technical Motivation	Implementation Mechanism
Minimize mistakes	Hide complexity / combine things	Default settings, templates
Increase velocity	Reduce friction	Automation
Improve products	Fill product gaps	New components
Enforce compliance	Restrict choice	Controls or wrappers
Reduce lock-in	Shield complexity	Wrappers / abstraction layers

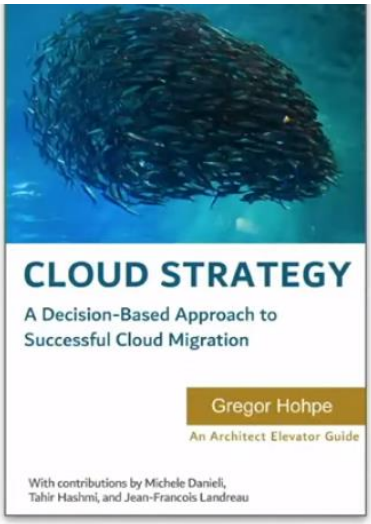
Fruit Salad or Fruit Basket

What does your platform look like? Is it a collection of independent services/capabilities or is it more like a fruit salad with different small services/capabilities that can be consumed together?

Caution: Failure does not respect abstraction



Hidden complexities often come out to face the users when problems arise, you need to empower them to be able to deal with abstracted use-cases.



CloudStrategyBook.com

Leanpub.com/platformstrategy