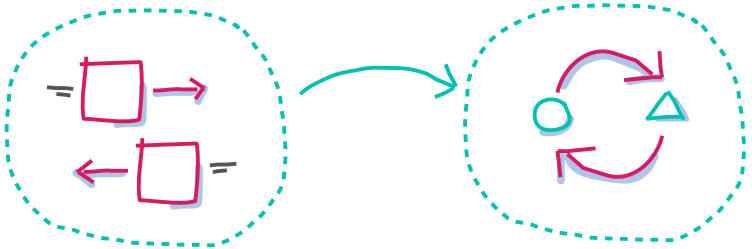
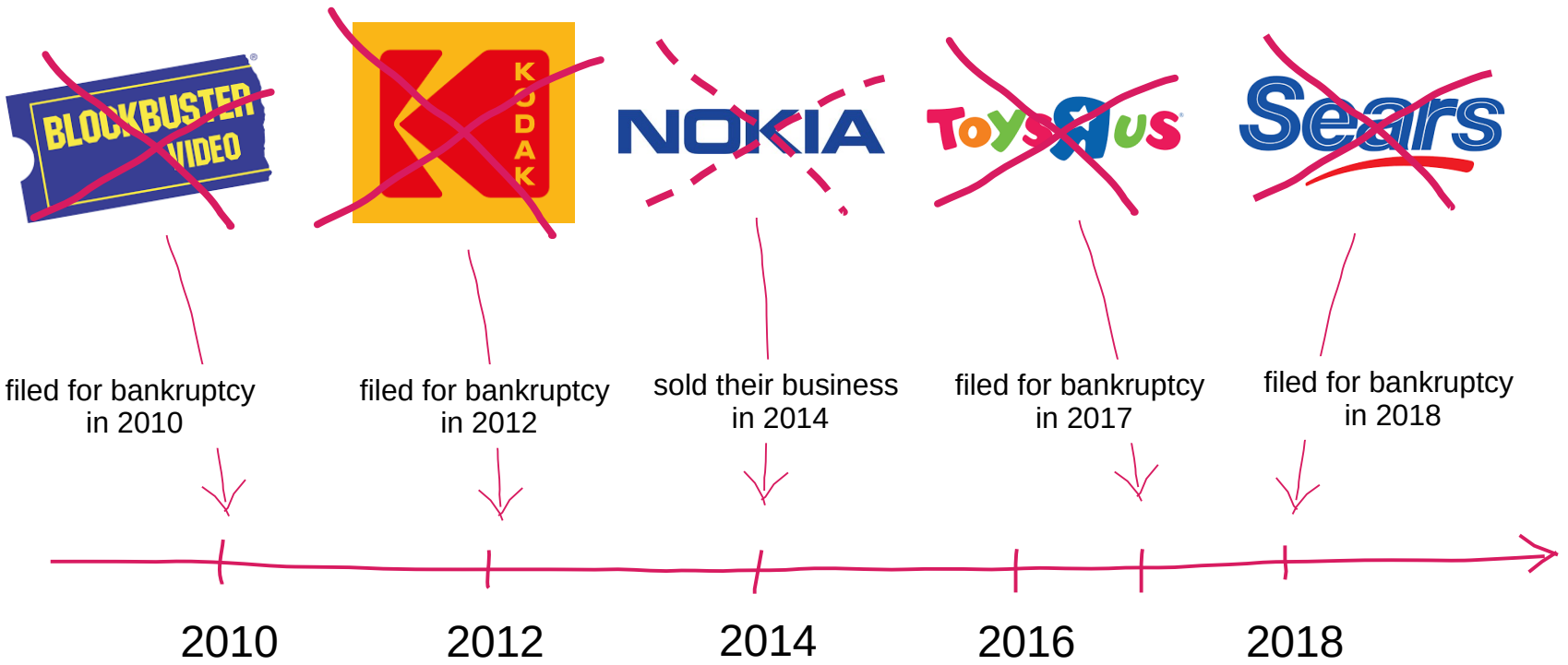


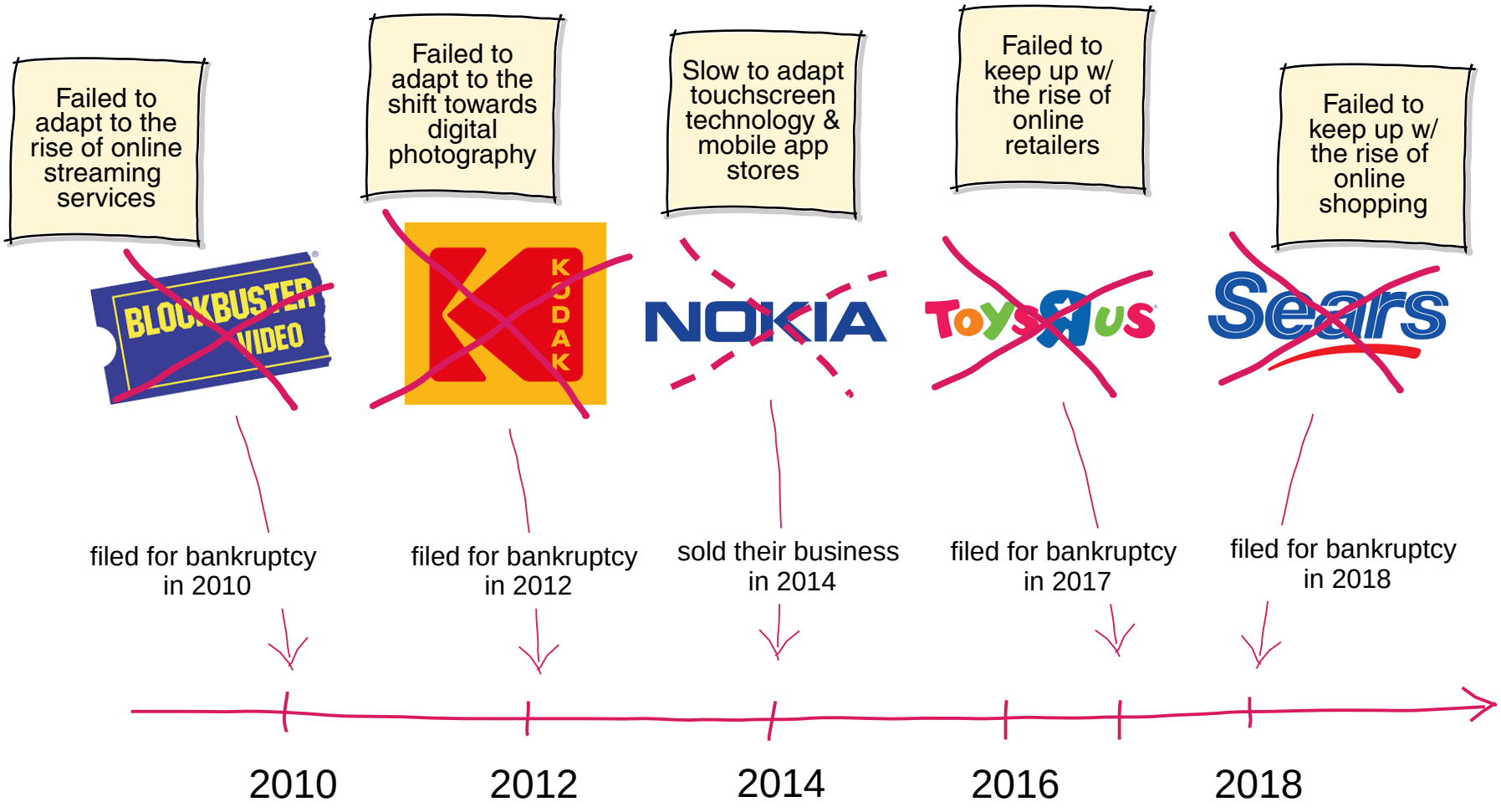
Adaptive Socio-Technical Systems w/ Architecture for Flow

Susanne Kaiser
Independent Tech Consultant
@suksr

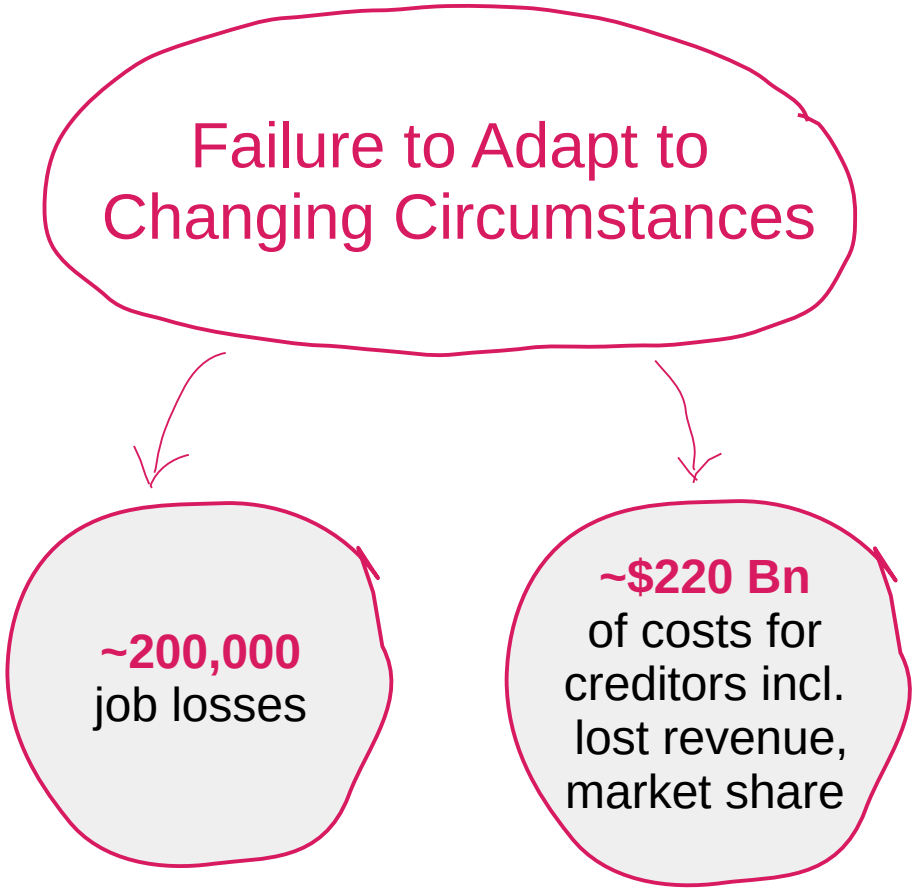
What do these companies have in common ?

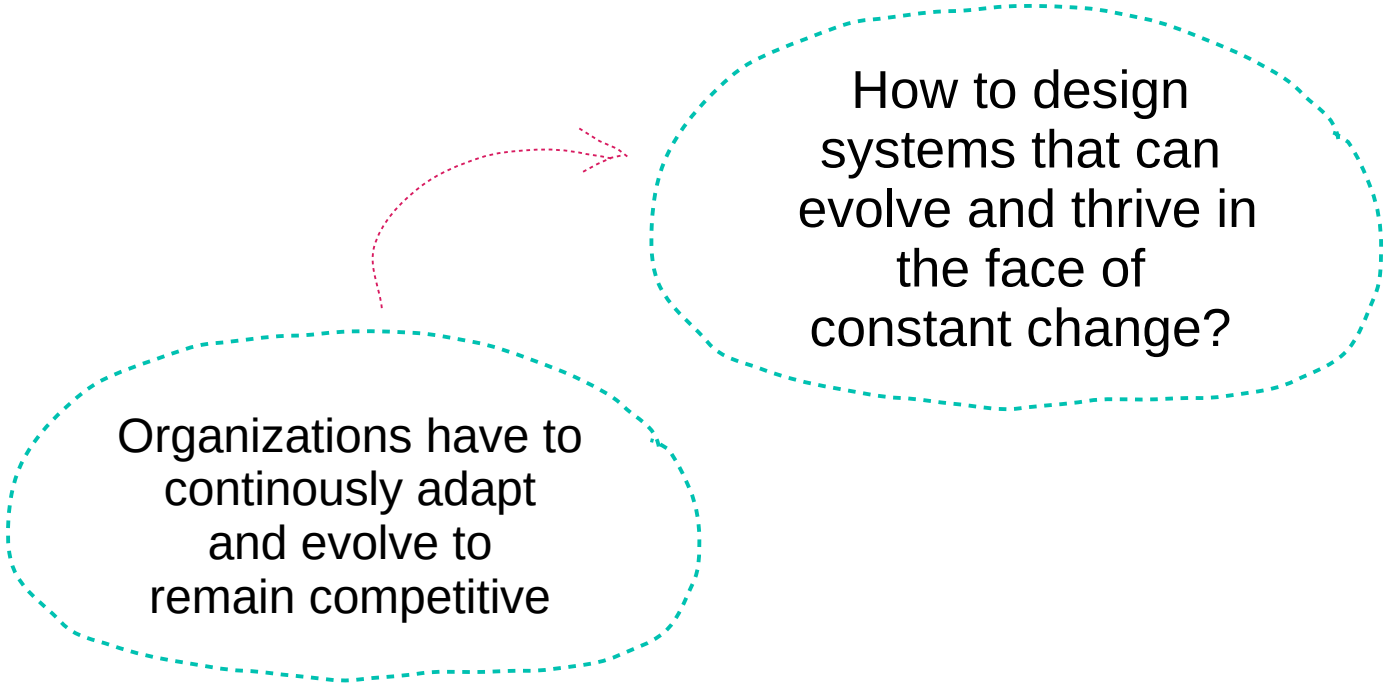
BLOCKBUSTER
VIDEO
KODAK
NOKIA
Toys"R"Us
Sears

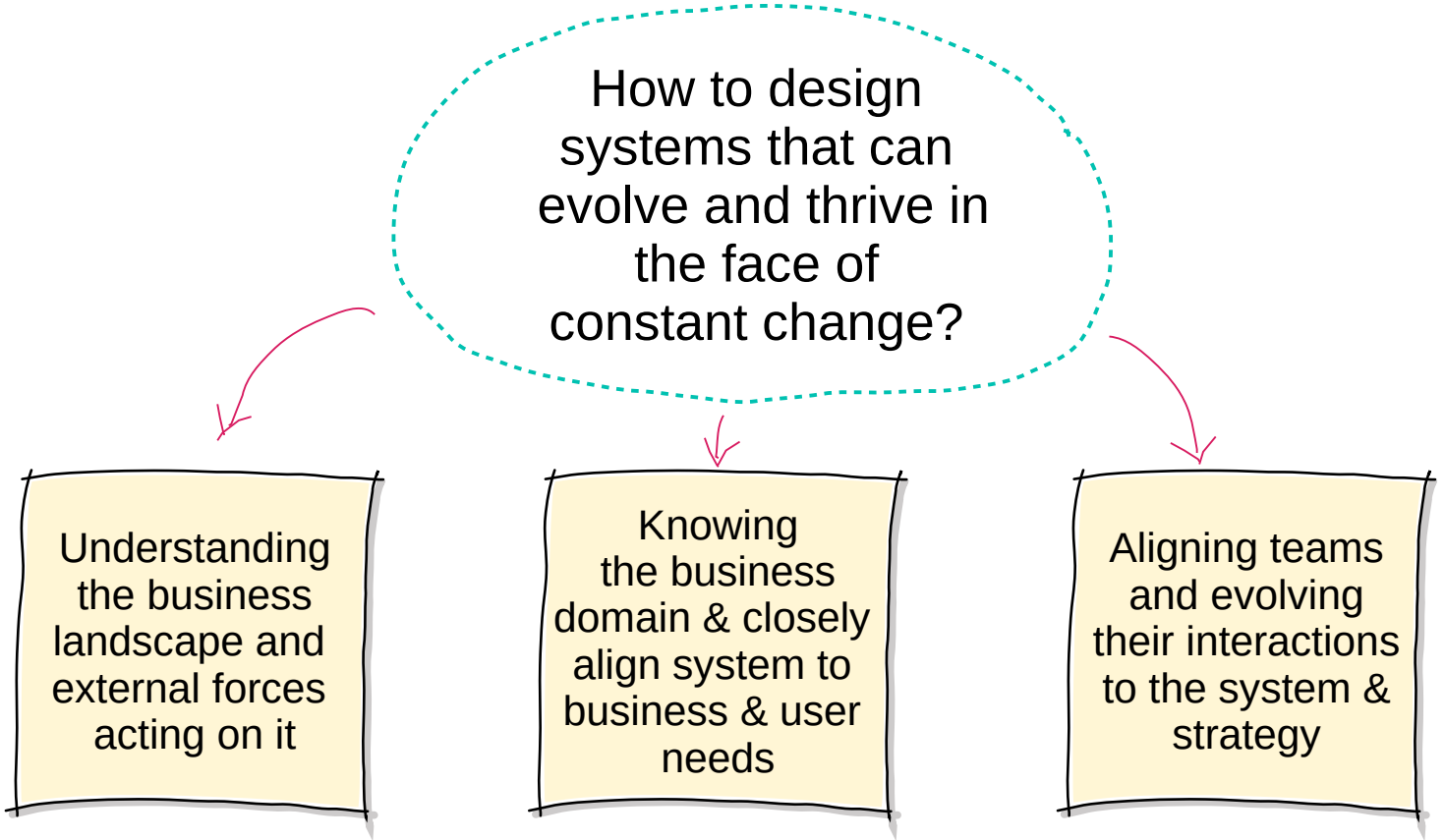

BLOCKBUSTER VIDEO
KODAK
NOKIA
Toys "R" Us
Sears
filed for bankruptcy in 2010
filed for bankruptcy in 2012
sold their business in 2014
filed for bankruptcy in 2017
filed for bankruptcy in 2018
2010
2012
2014
2016
2018

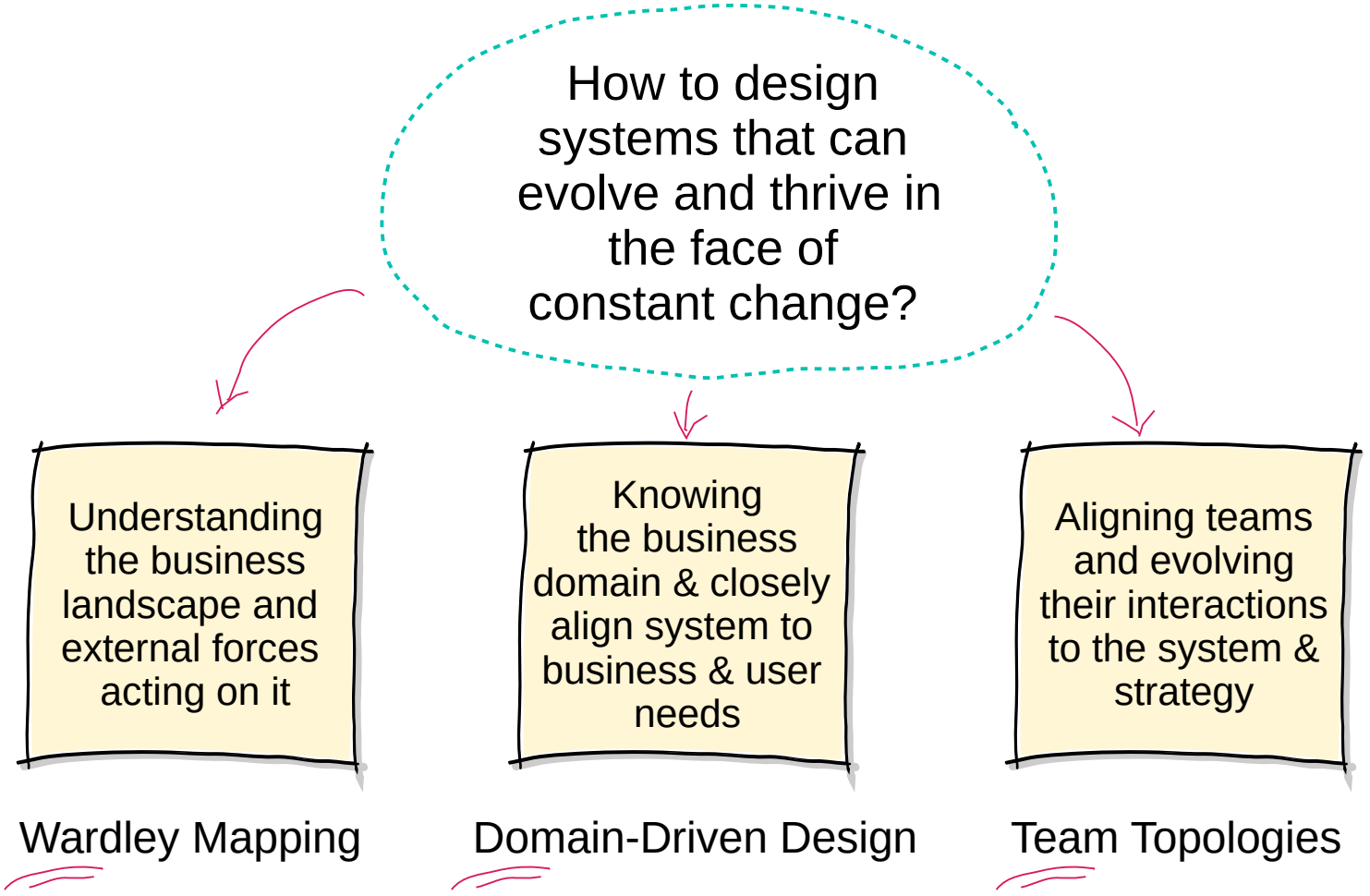

Failed to adapt to the rise of online streaming services
Failed to adapt to the shift towards digital photography
Slow to adapt touchscreen technology & mobile app stores
Failed to keep up w/ the rise of online retailers
Failed to keep up w/ the rise of online shopping
BLOCKBUSTER VIDEO
KODAK
NOKIA
Toys "R" Us
Sears
filed for bankruptcy in 2010
filed for bankruptcy in 2012
sold their business in 2014
filed for bankruptcy in 2017
filed for bankruptcy in 2018
2010
2012
2014
2016
2018

@suksr


Failure to Adapt to Changing Circumstances
~200,000
job losses
~$220 Bn
of costs for creditors incl. lost revenue, market share


How to design systems that can evolve and thrive in the face of constant change?
Organizations have to continously adapt and evolve to remain competitive


How to design systems that can evolve and thrive in the face of constant change?
Understanding the business landscape and external forces acting on it
Knowing the business domain & closely align system to business & user needs
Aligning teams and evolving their interactions to the system & strategy

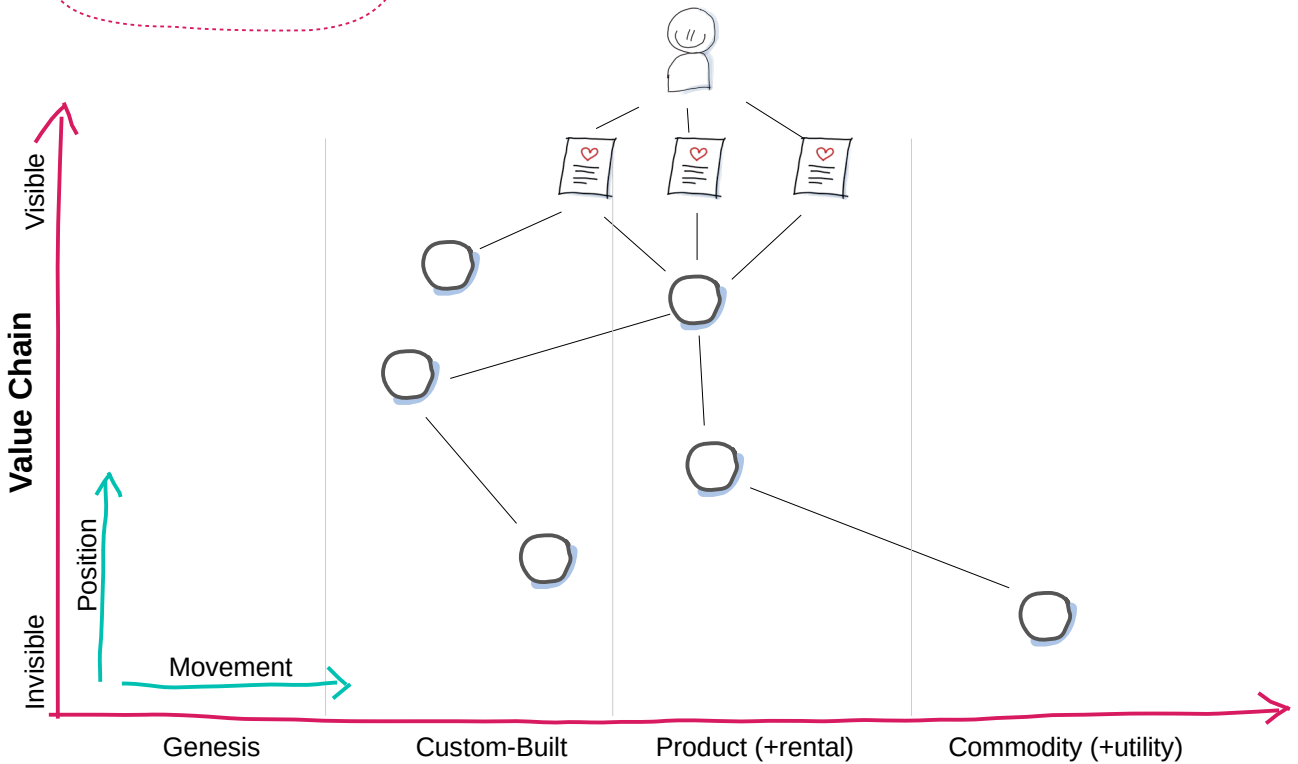

How to design systems that can evolve and thrive in the face of constant change?
Understanding the business landscape and external forces acting on it
Knowing the business domain & closely align system to business & user needs
Aligning teams and evolving their interactions to the system & strategy
Wardley Mapping
Domain-Driven Design
Team Topologies

Evolving a Legacy System

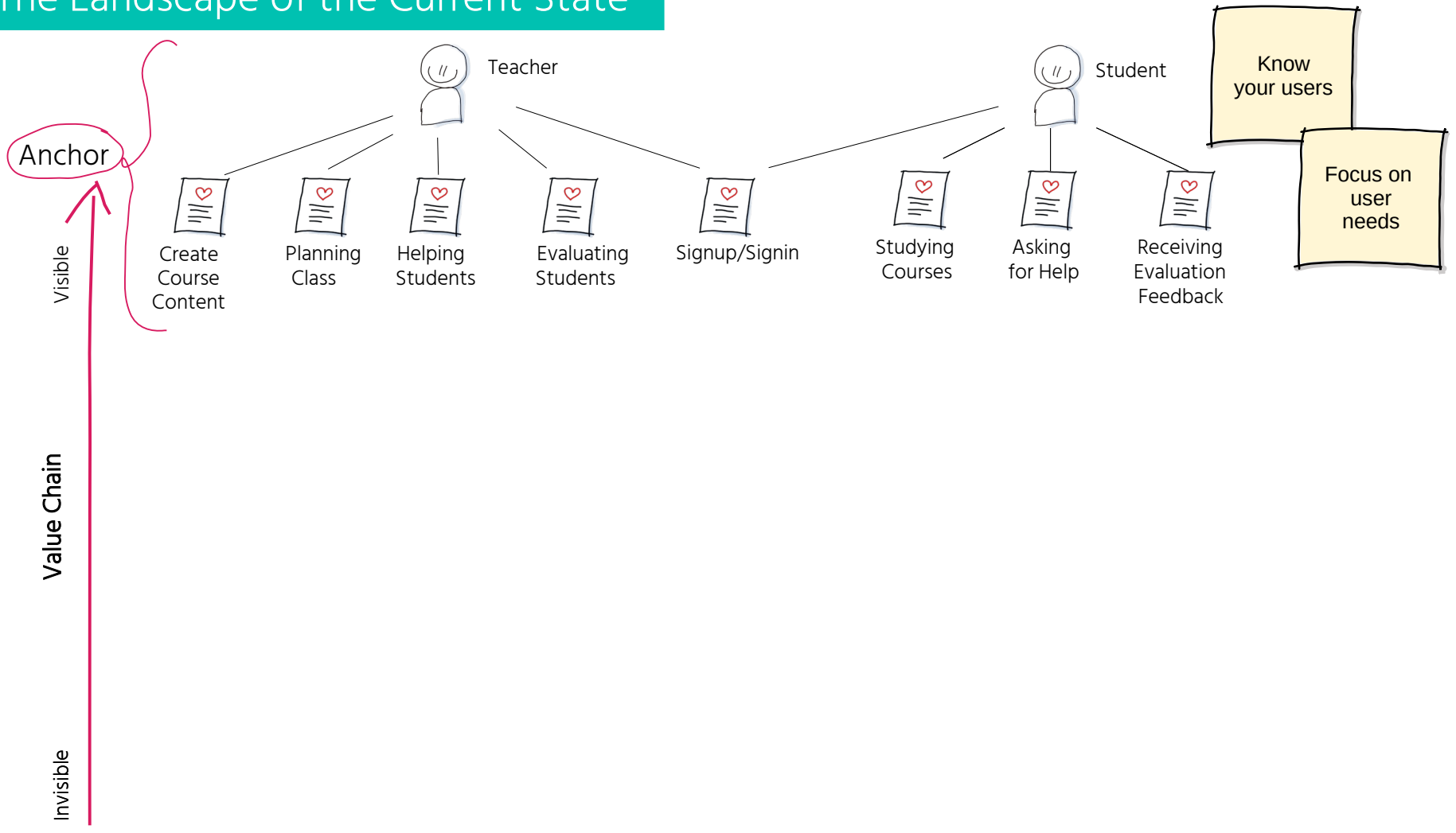
Image source: https://www.food-management.com

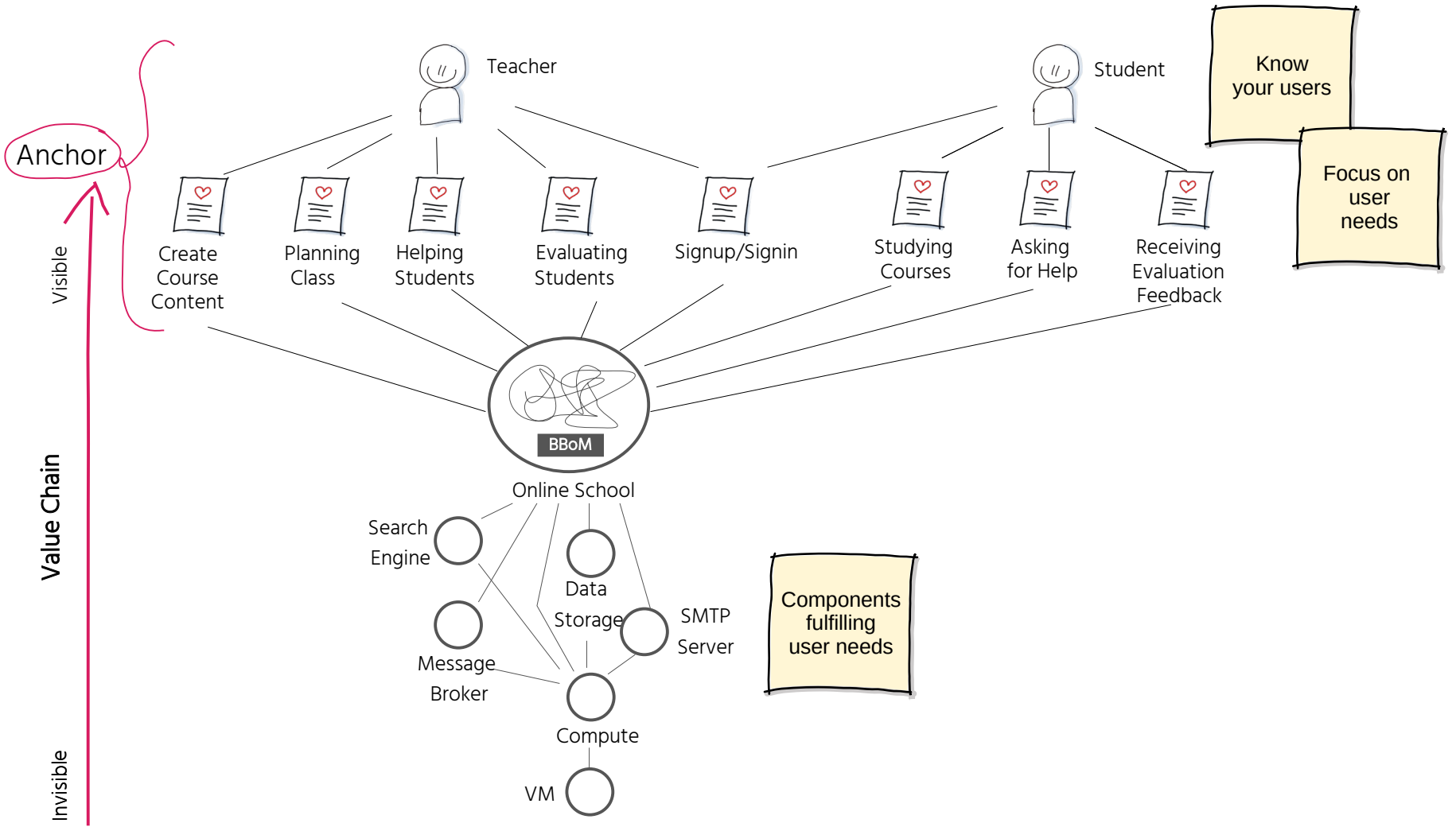
Understanding the business landscape

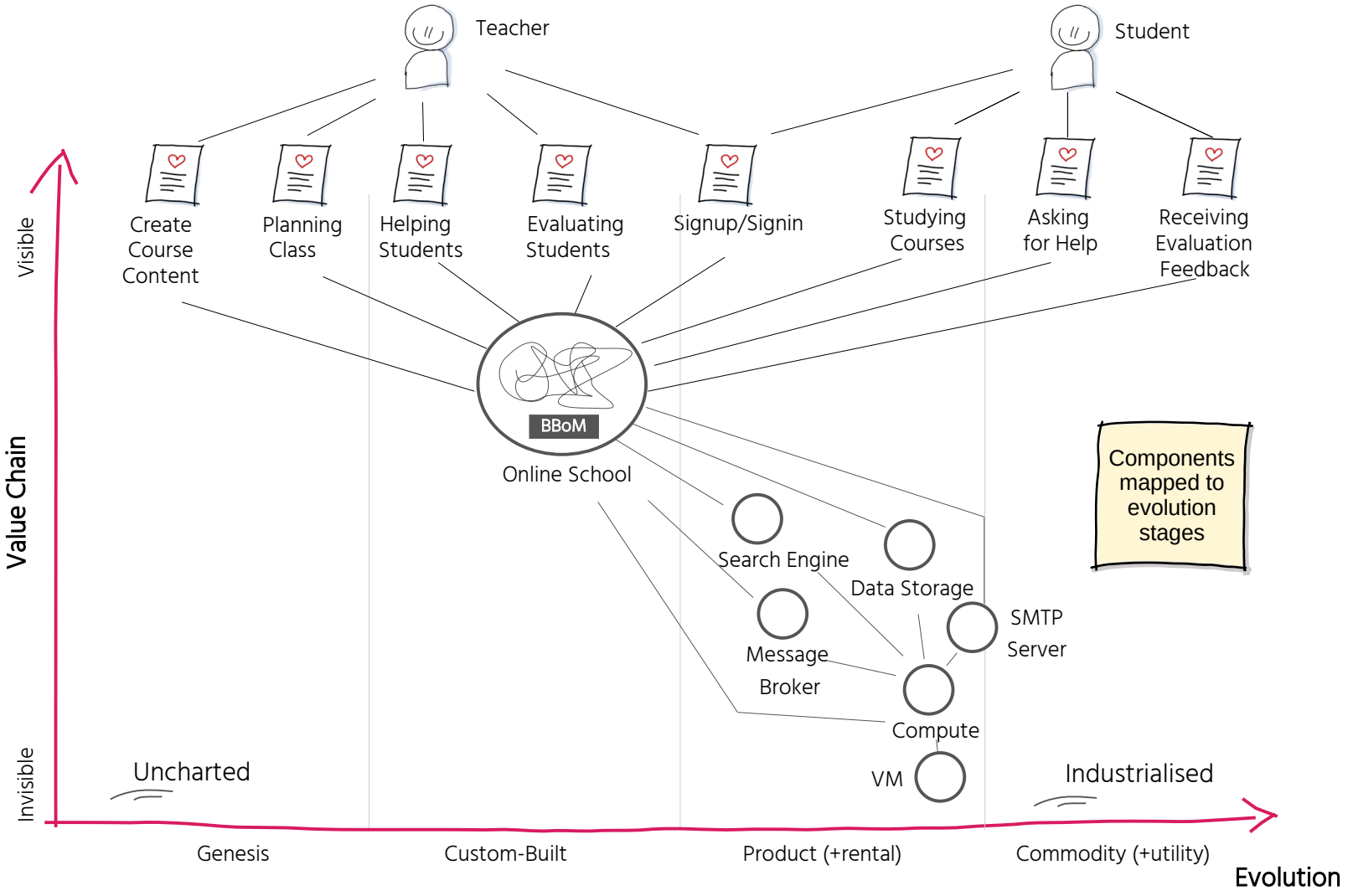
Wardley Map

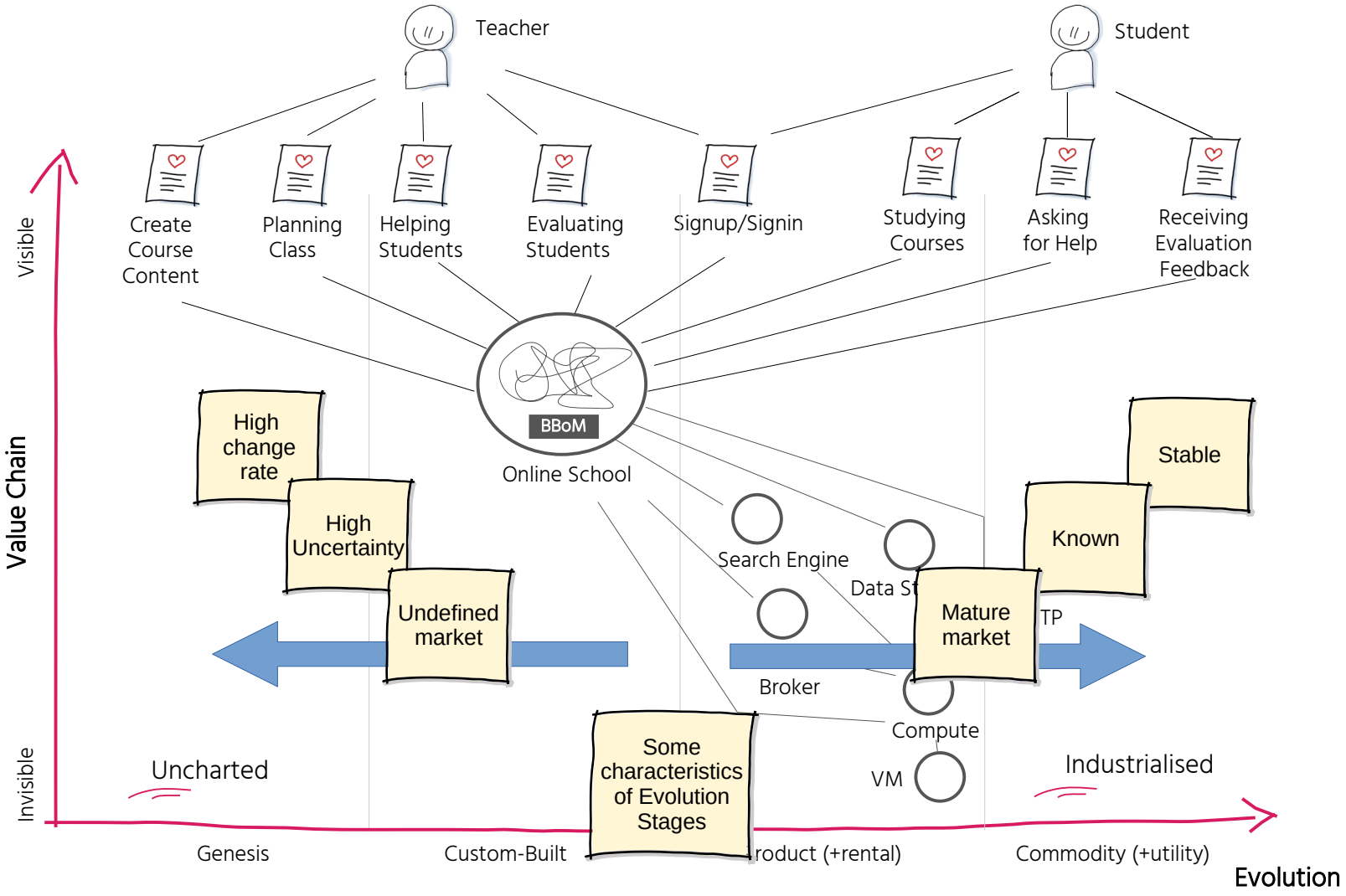


The Landscape of the Current State

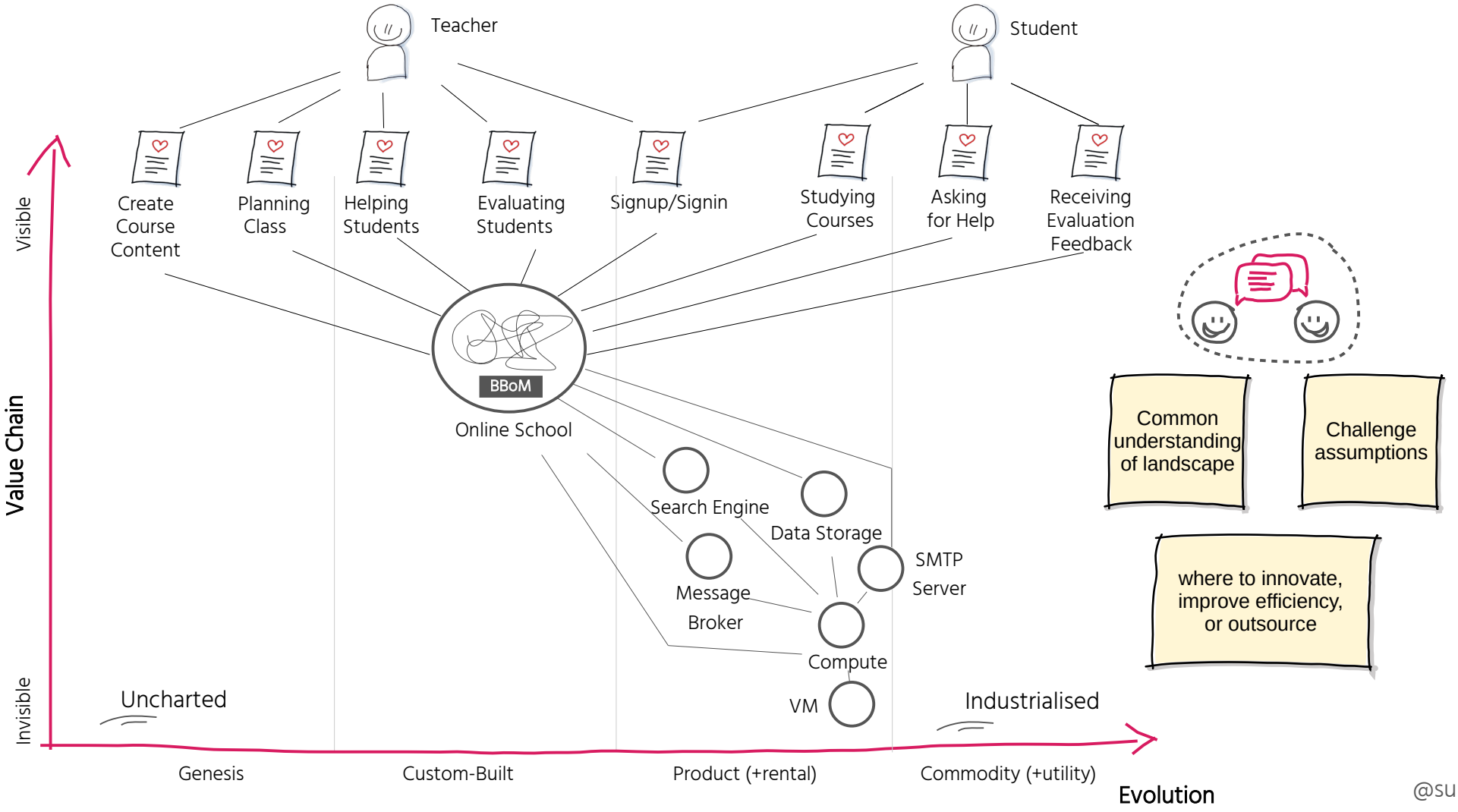



Anchor
Visible
Value Chain
Invisible
Teacher
Student
Know your users
Focus on user needs
Create Course Content
Planning Class
Helping Students
Evaluating Students
Signup/Signin
Studying Courses
Asking for Help
Receiving Evaluation Feedback
BBoM
Online School
Search Engine
Data Storage
SMTP Server
Message Broker
Compute
VM
Components fulfilling user needs


Teacher
Student
Create Course Content
Planning Class
Helping Students
Evaluating Students
Signup/Signin
Studying Courses
Asking for Help
Receiving Evaluation Feedback
BBoM
Online School
Components mapped to evolution stages
Search Engine
Data Storage
SMTP Server
Message Broker
Compute
VM
Visible
Invisible
Value Chain
Uncharted
Industrialised
Genesis
Custom-Built
Product (+rental)
Commodity (+utility)
Evolution

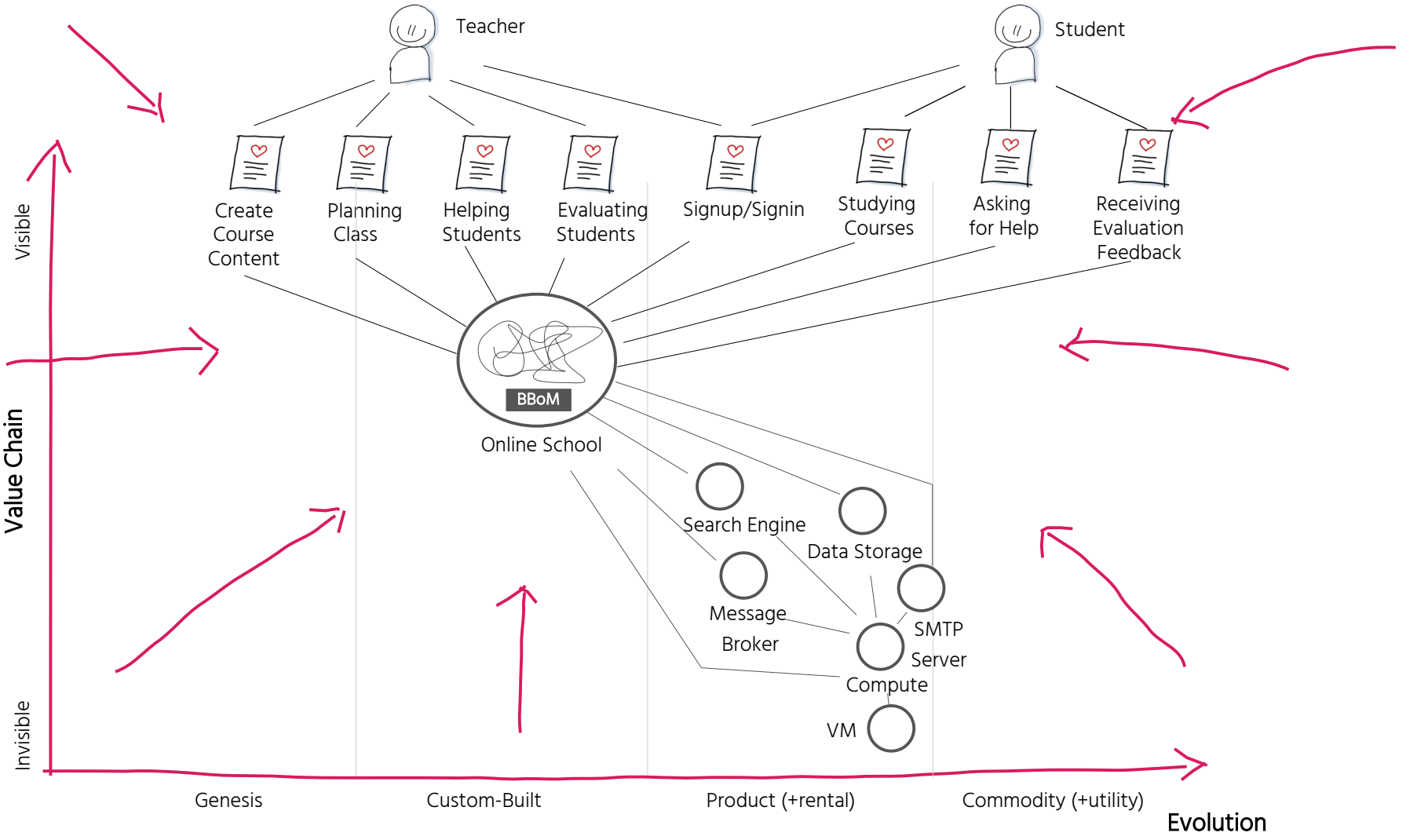

Teacher
Student
Create Course Content
Planning Class
Helping Students
Evaluating Students
Signup/Signin
Studying Courses
Asking for Help
Receiving Evaluation Feedback
BBoM
Online School
Visible
Invisible
Value Chain
High change rate
High Uncertainty
Undefined market
Search Engine
Data St
Broker
Mature market
TP
Stable
Known
Compute
VM
Some characteristics of Evolution Stages
Uncharted
Industrialised
Genesis
Custom-Built
roduct (+rental)
Commodity (+utility)
Evolution

A Wardley Map of the Current State

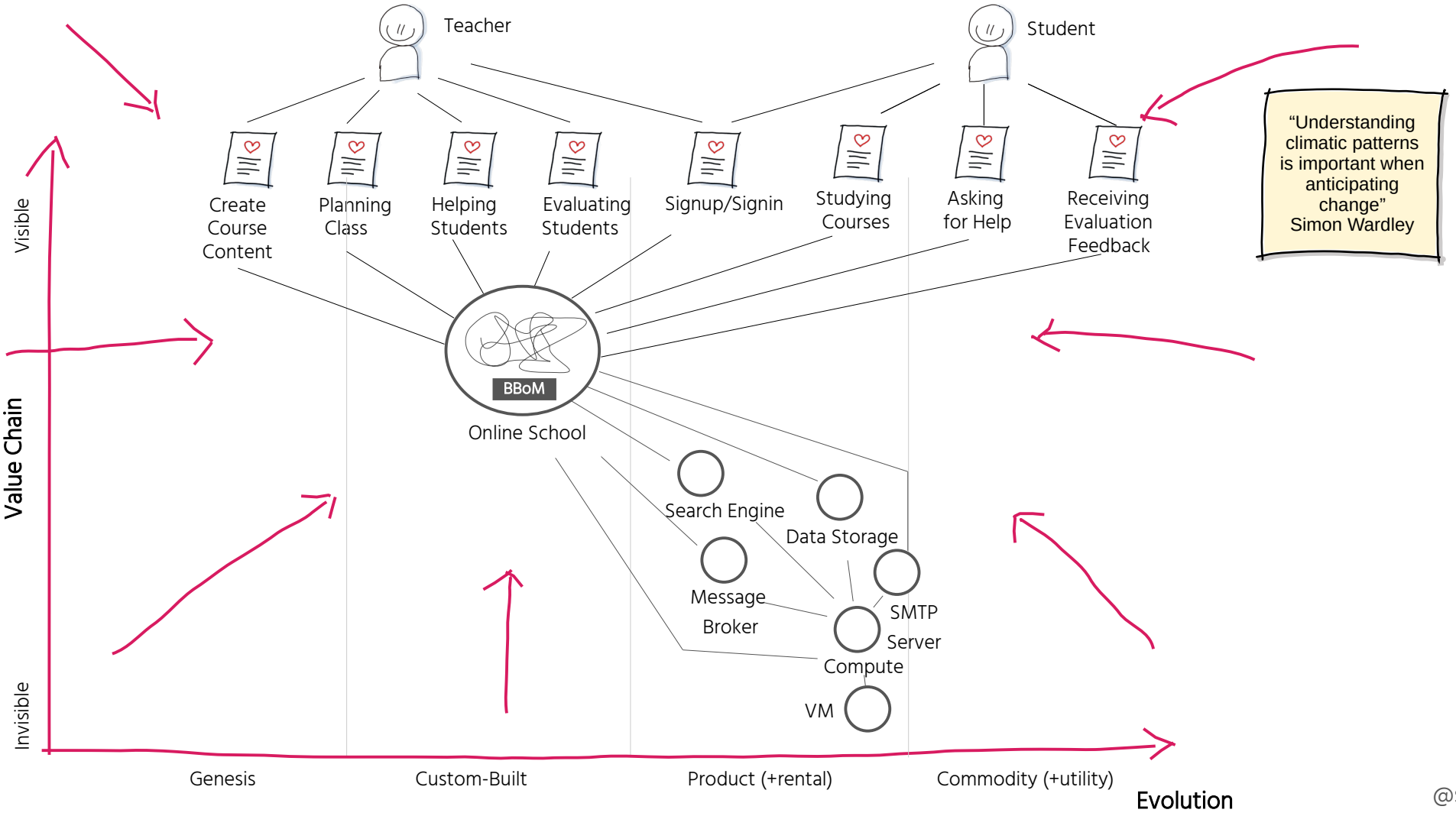


@suksr

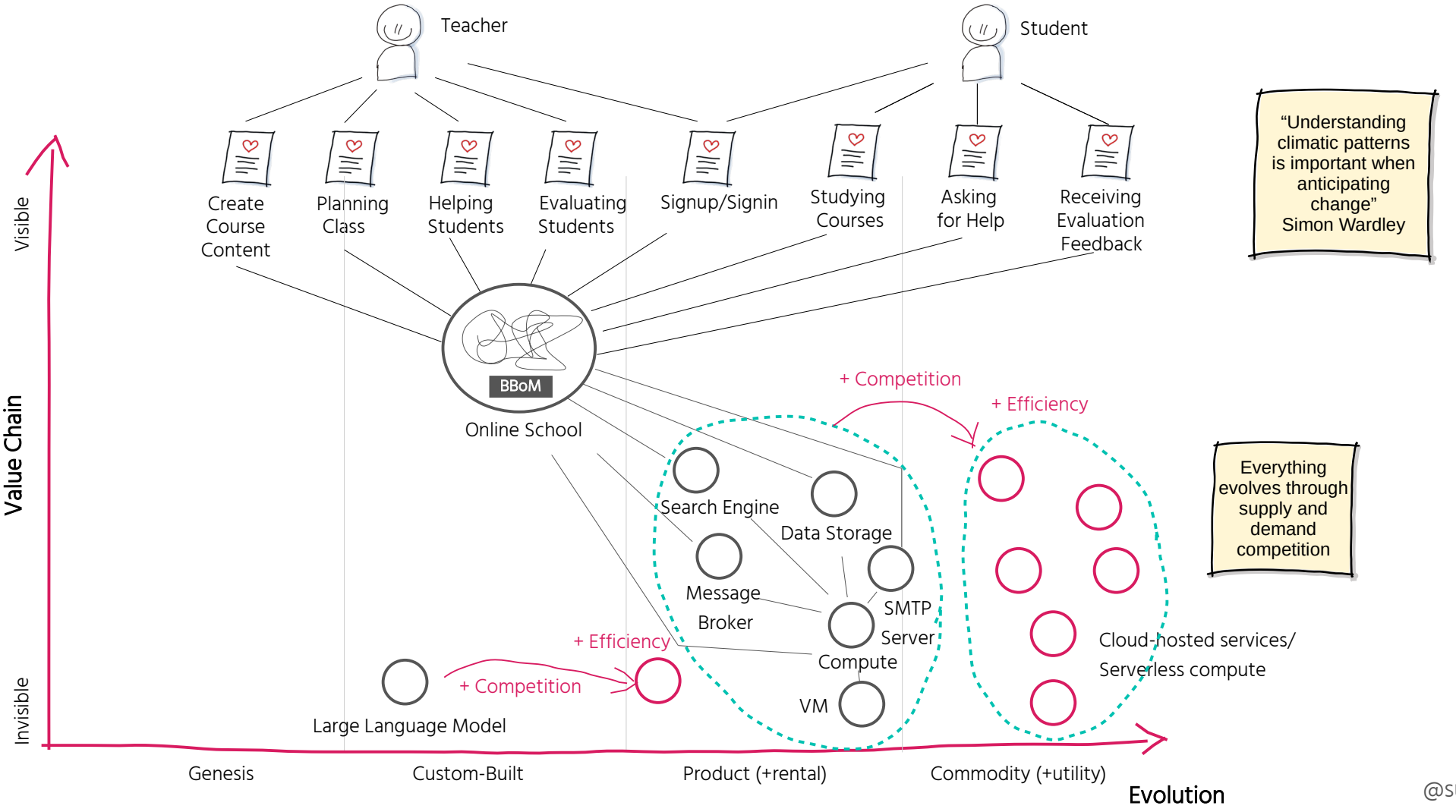
External Forces Impacting the Landscape: Climatic Patterns



External Forces Impacting the Landscape: Climatic Patterns

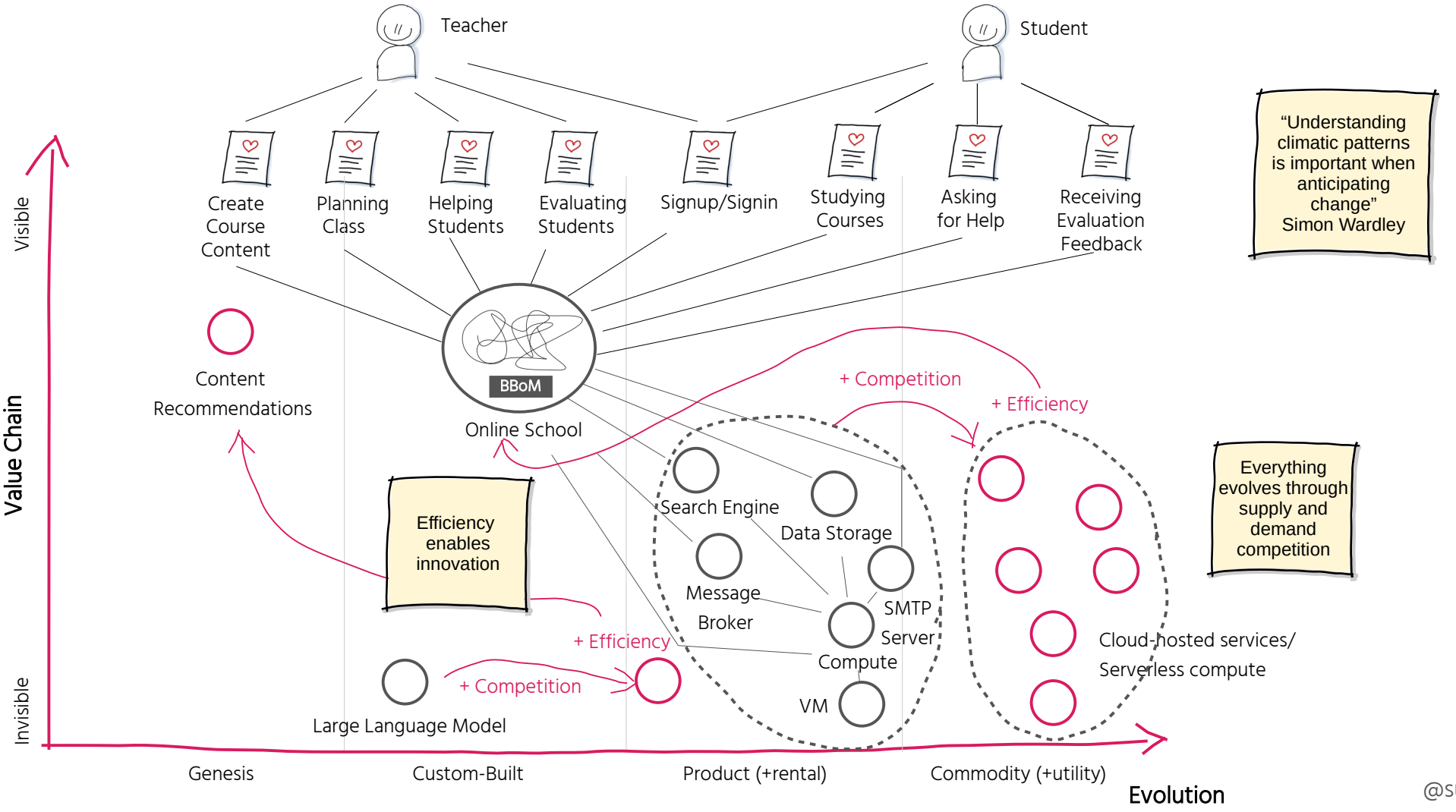


External Forces Impacting the Landscape: Climatic Patterns



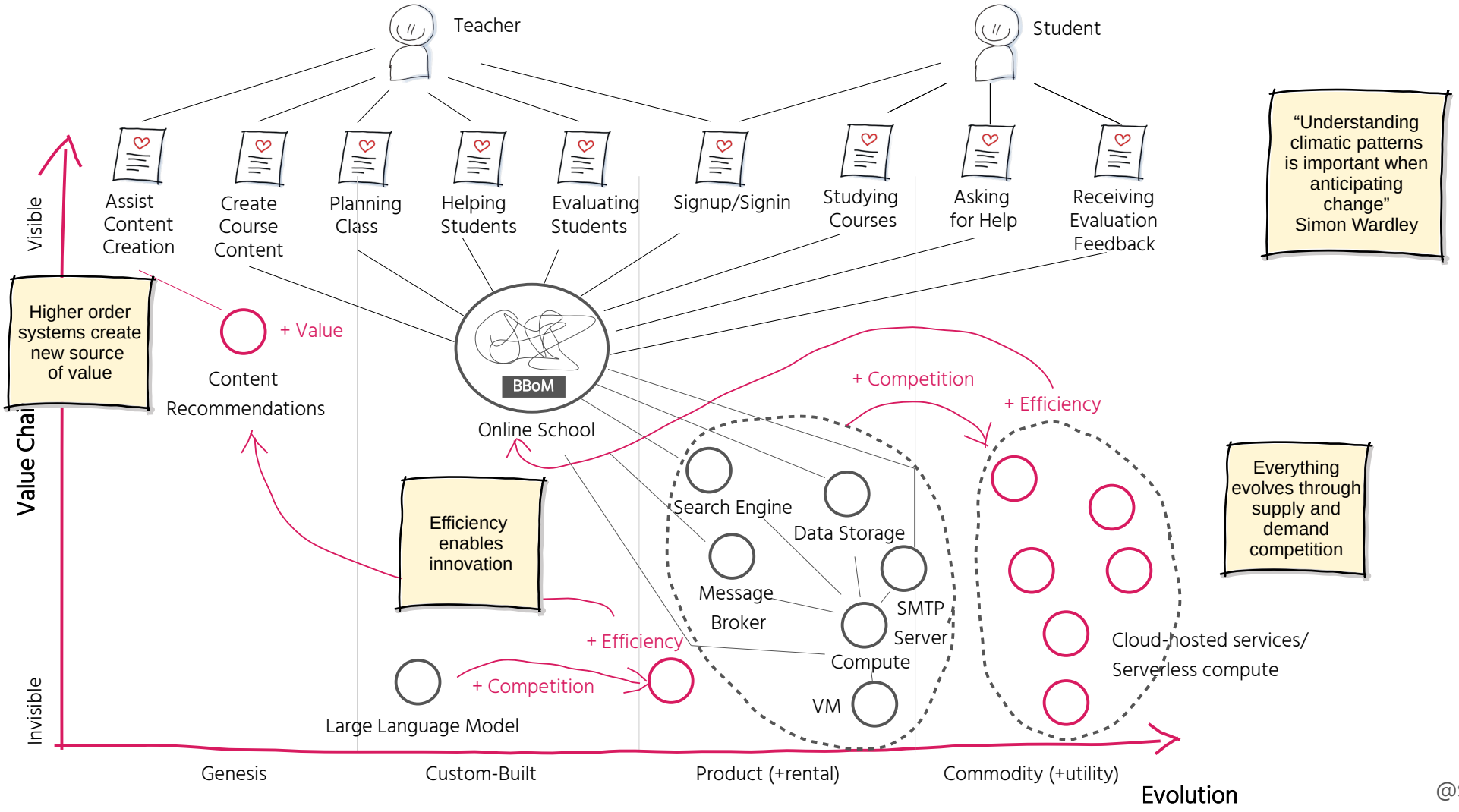
@suksr

External Forces Impacting the Landscape: Climatic Patterns



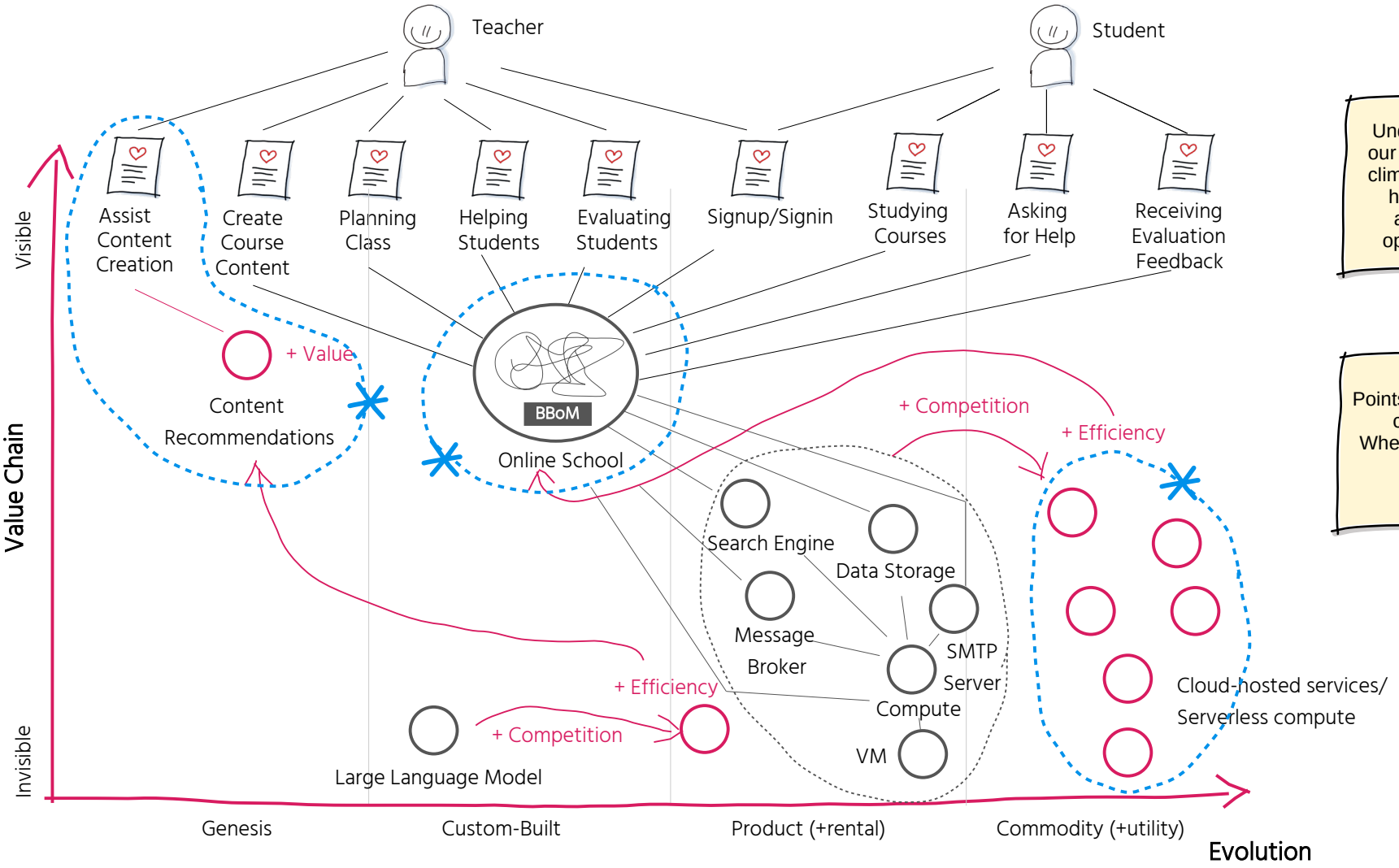
@suksr

External Forces Impacting the Landscape: Climatic Patterns



@suksr

Understanding Landscape & Climatic Patterns

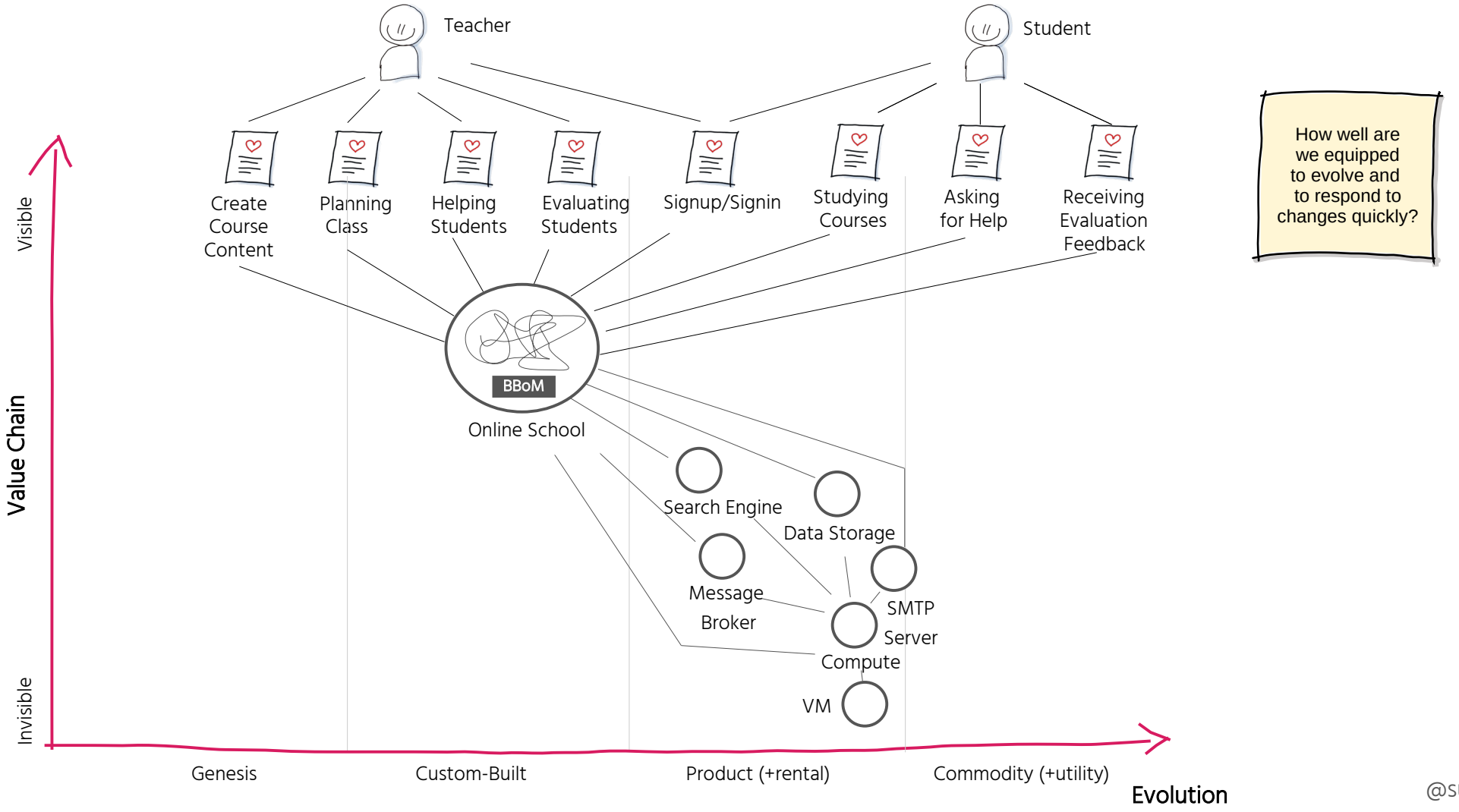


Understanding our landscape & climatic patterns helps us to anticipate opportunities

Points of potential change / Where to invest?

@suksr

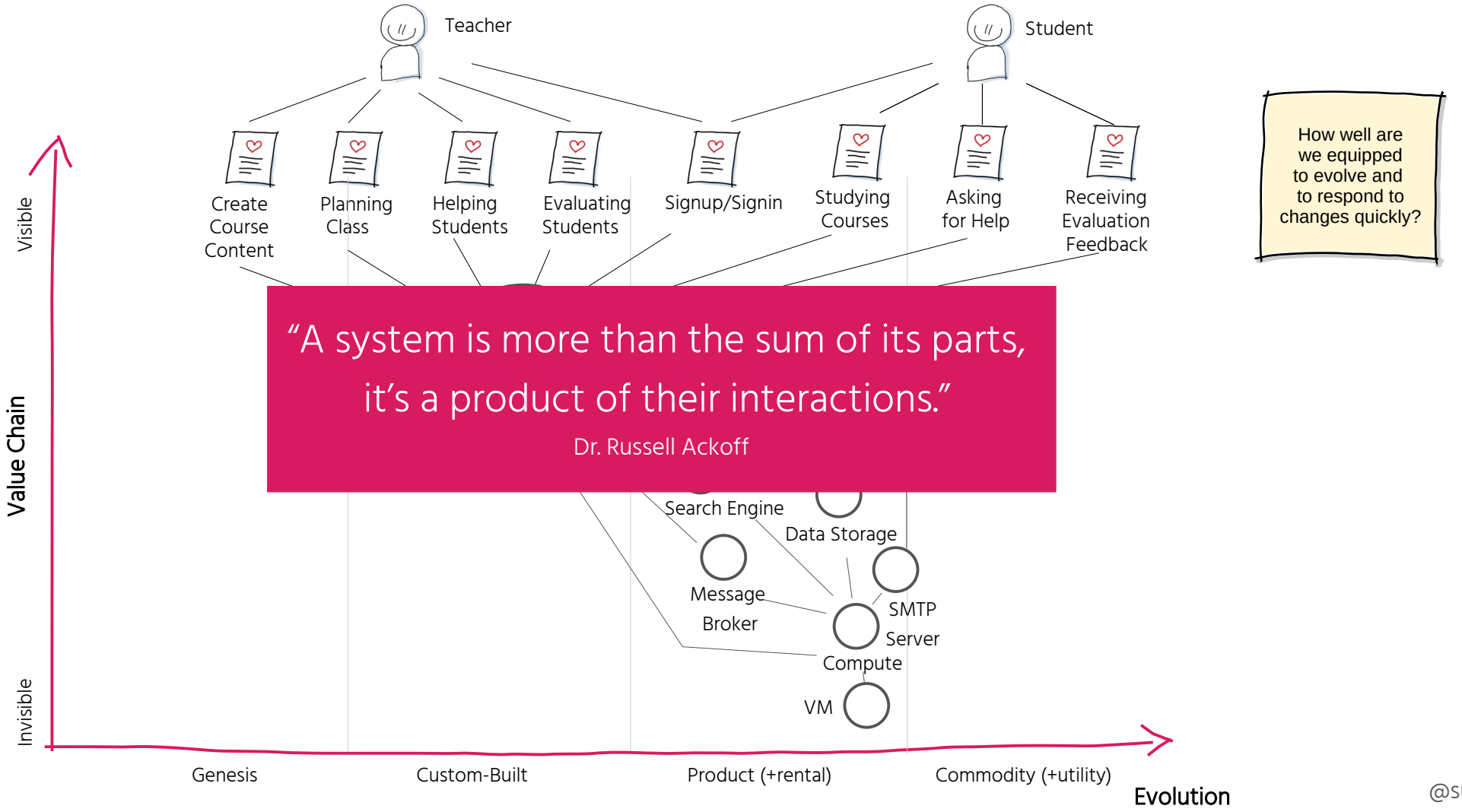
Assessing Responsiveness to Change



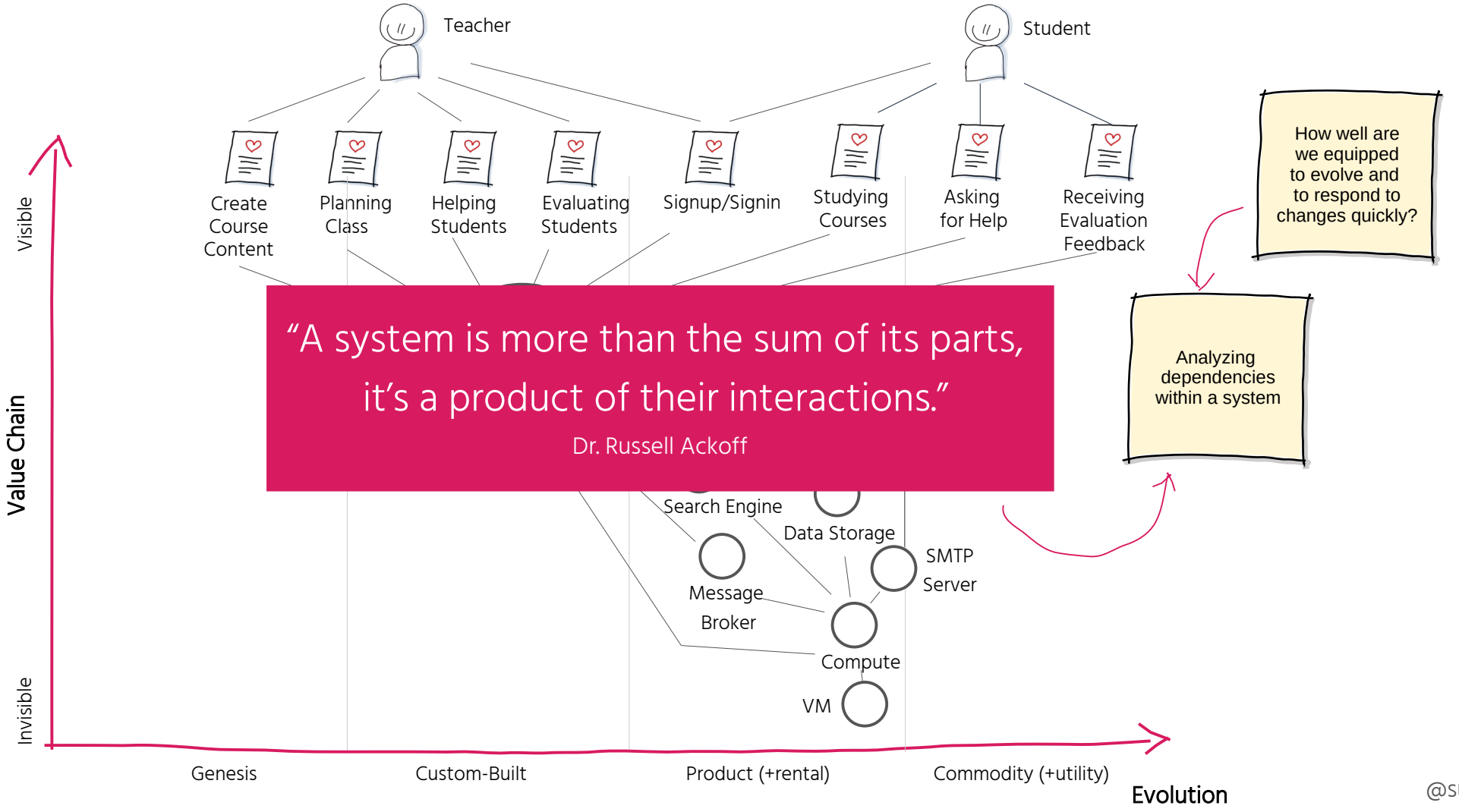
How well are we equipped to evolve and to respond to changes quickly?

@suksr

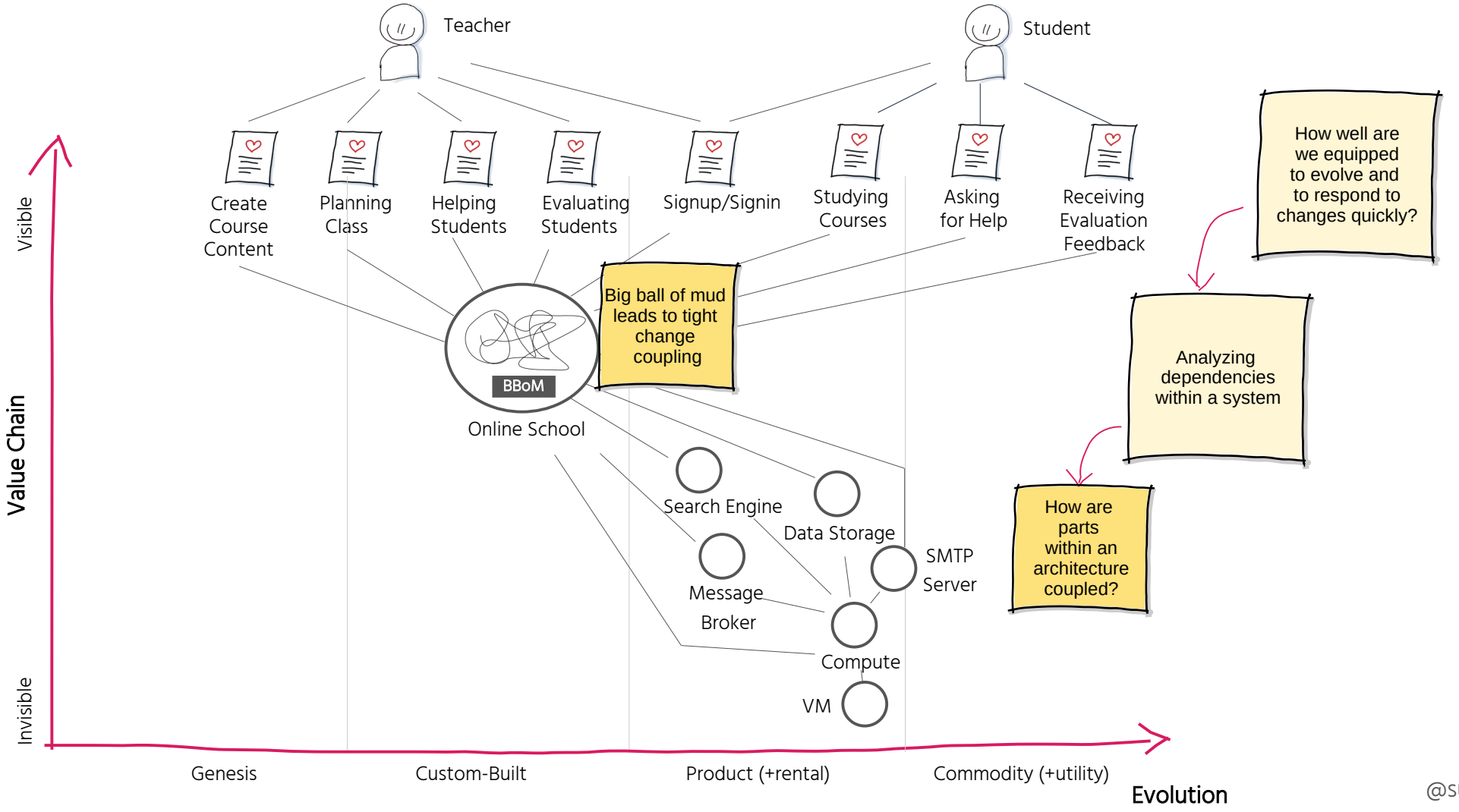
Assessing Responsiveness to Change



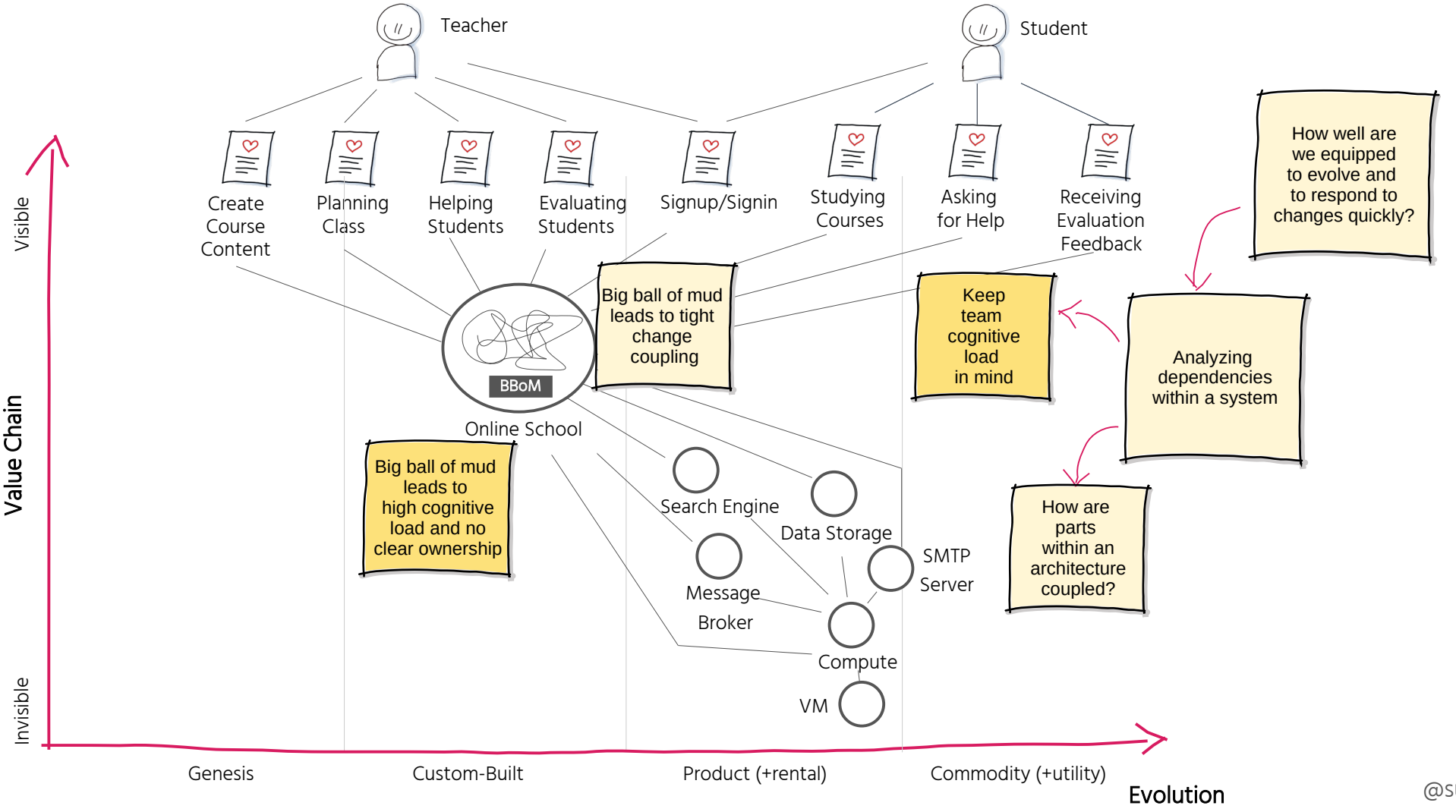
Assessing Responsiveness to Change



Assessing Responsiveness to Change

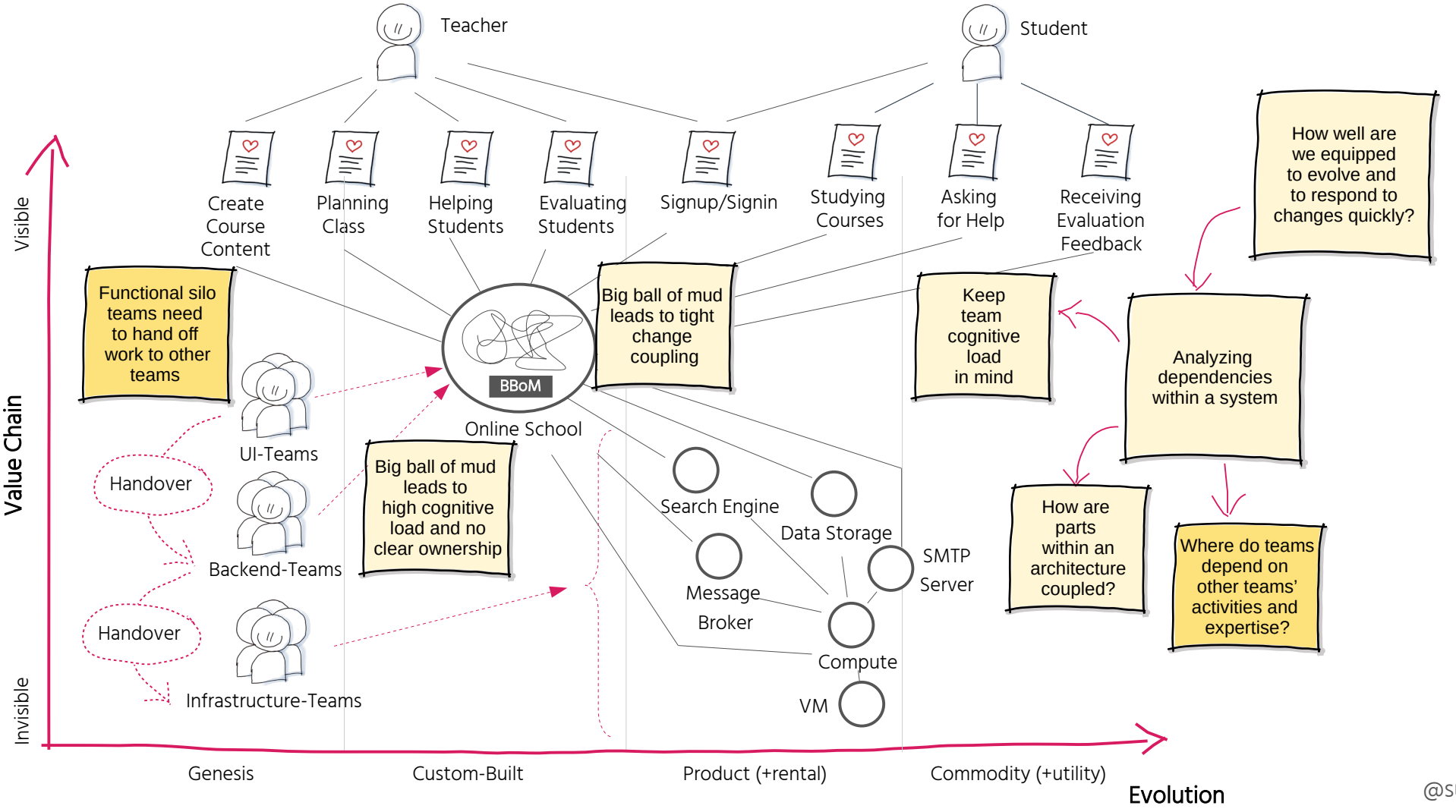


Assessing Responsiveness to Change



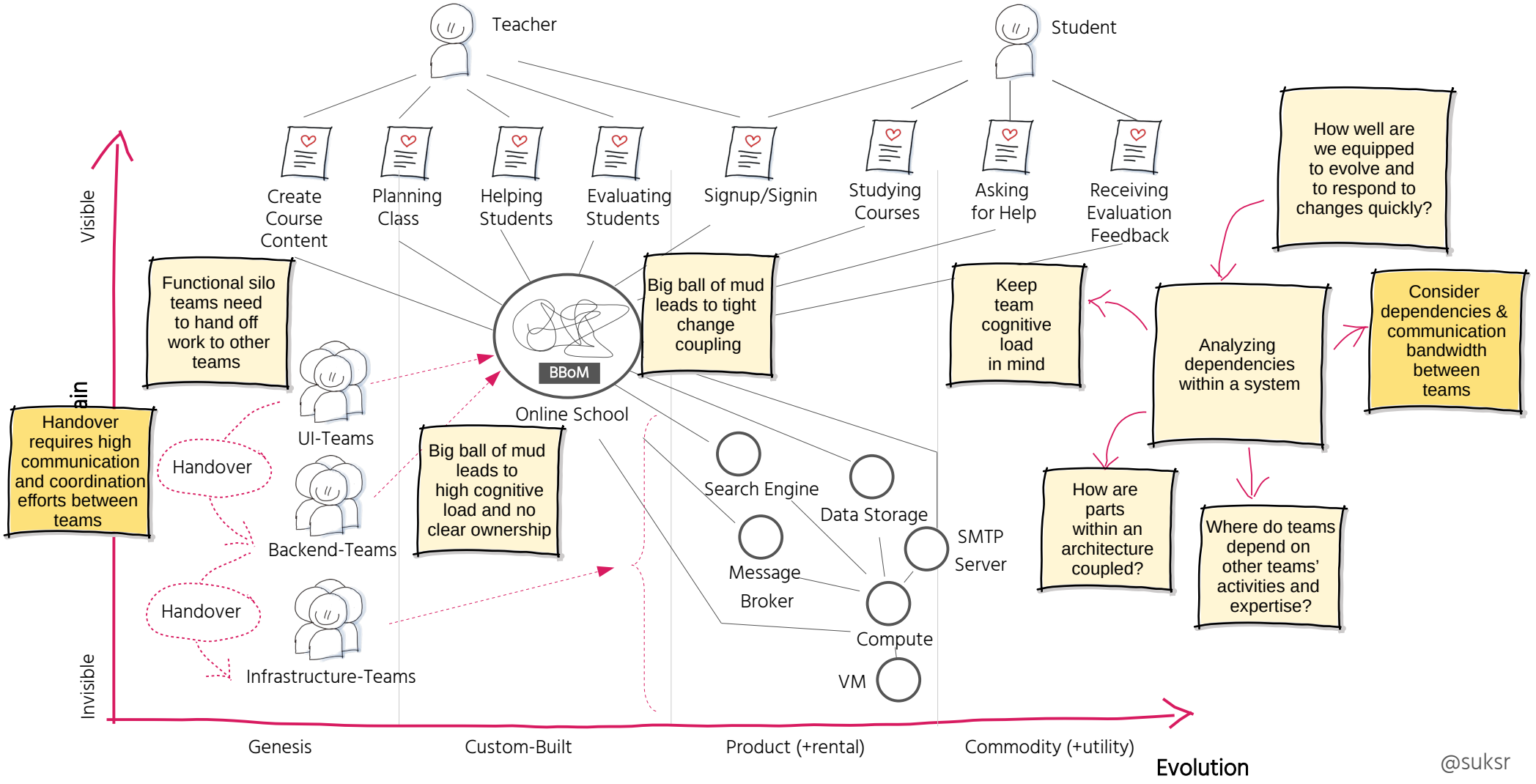
@suksr

Assessing Responsiveness to Change



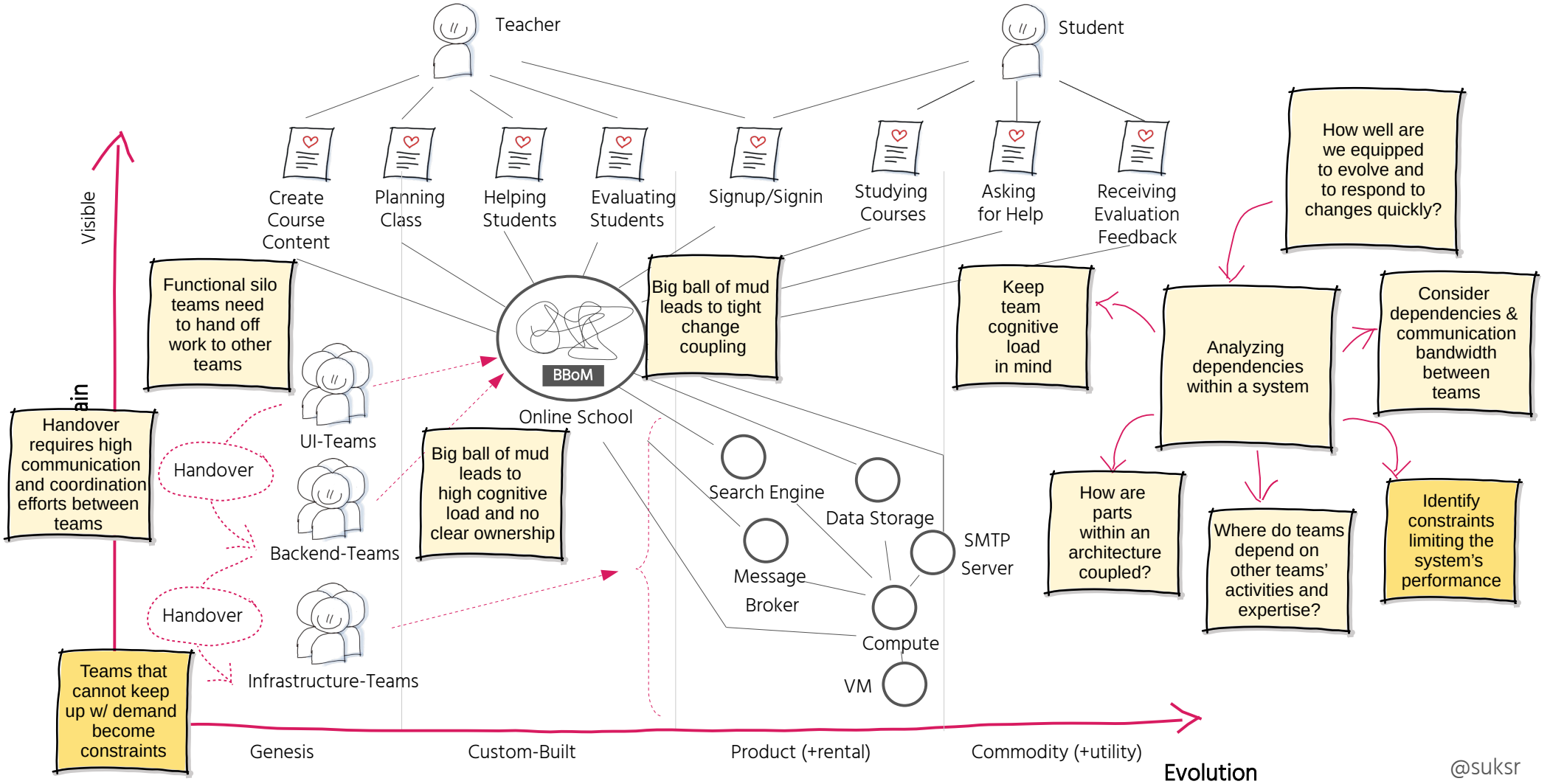
@suksr

Assessing Responsiveness to Change

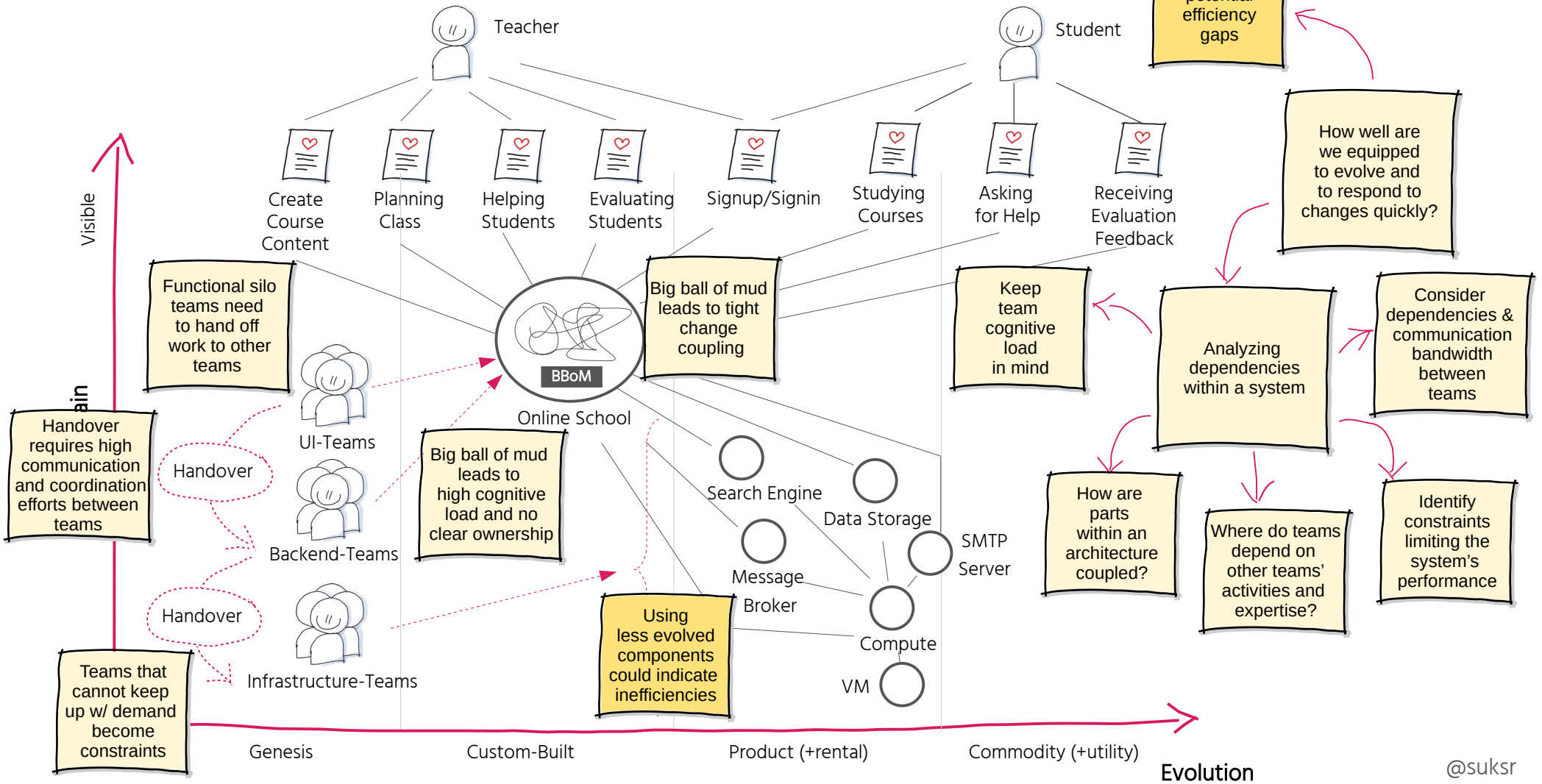


@suksr

Assessing Responsiveness to Change

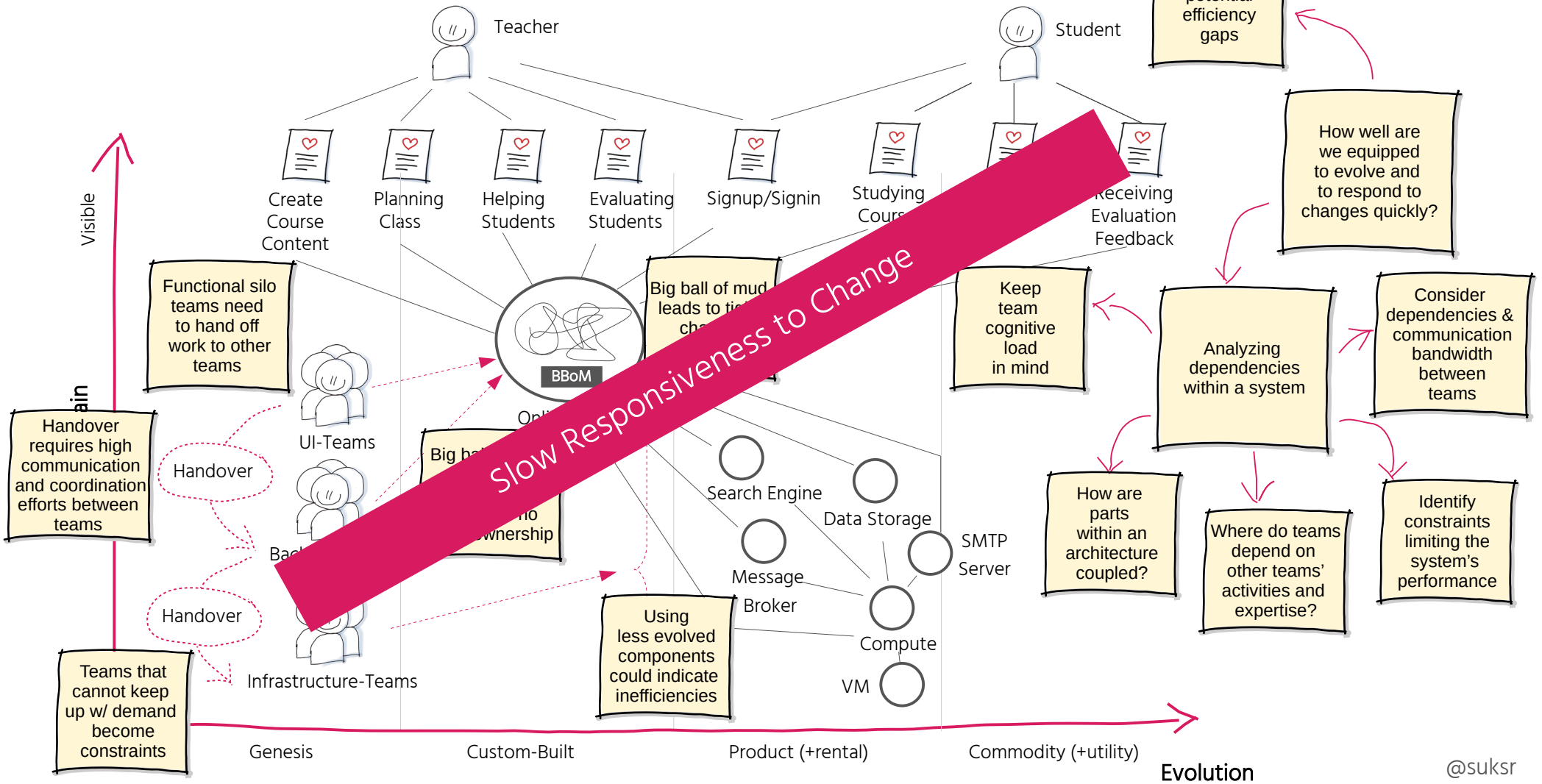


Assessing Responsiveness to Change



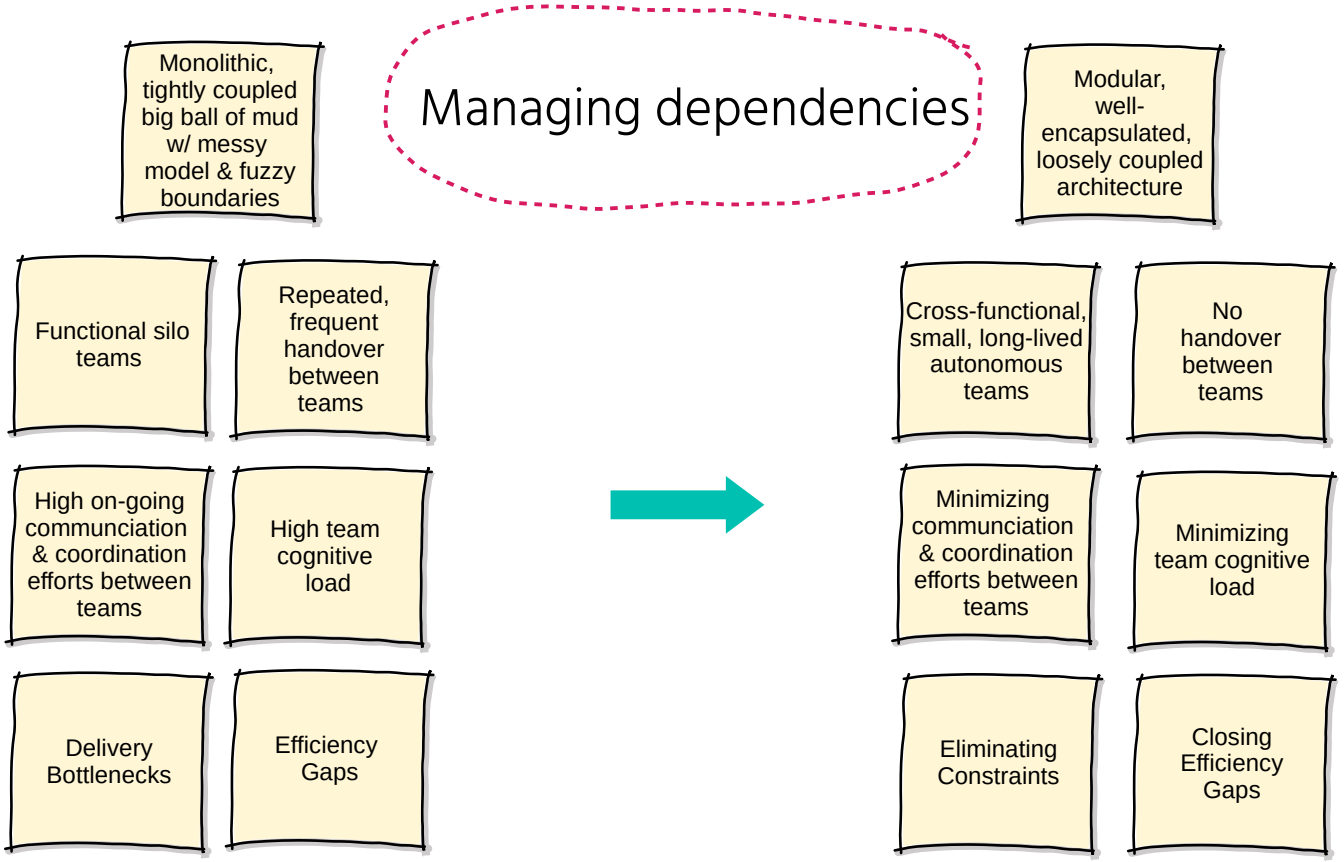
@suksr

Assessing Responsiveness to Change

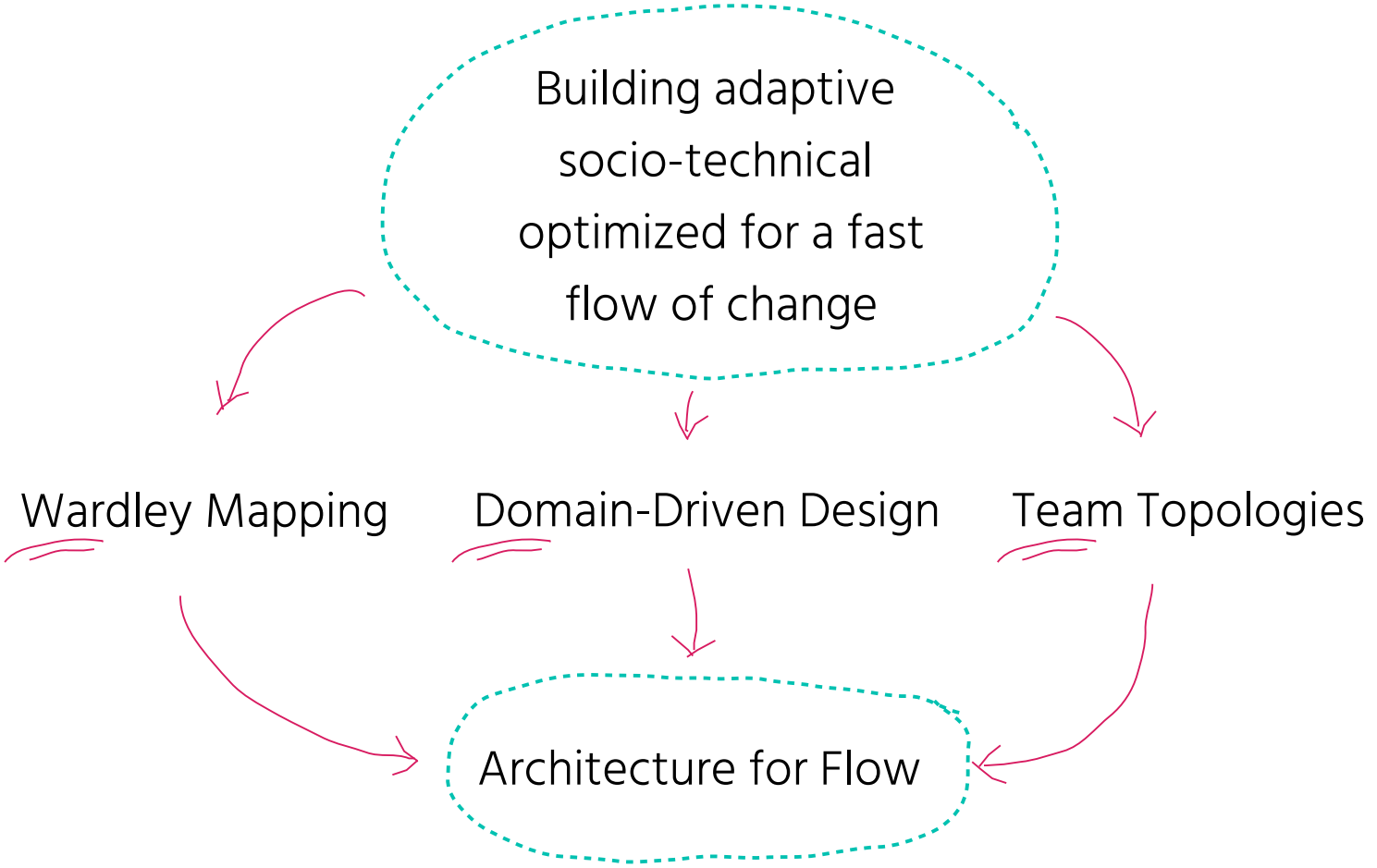


@suksr

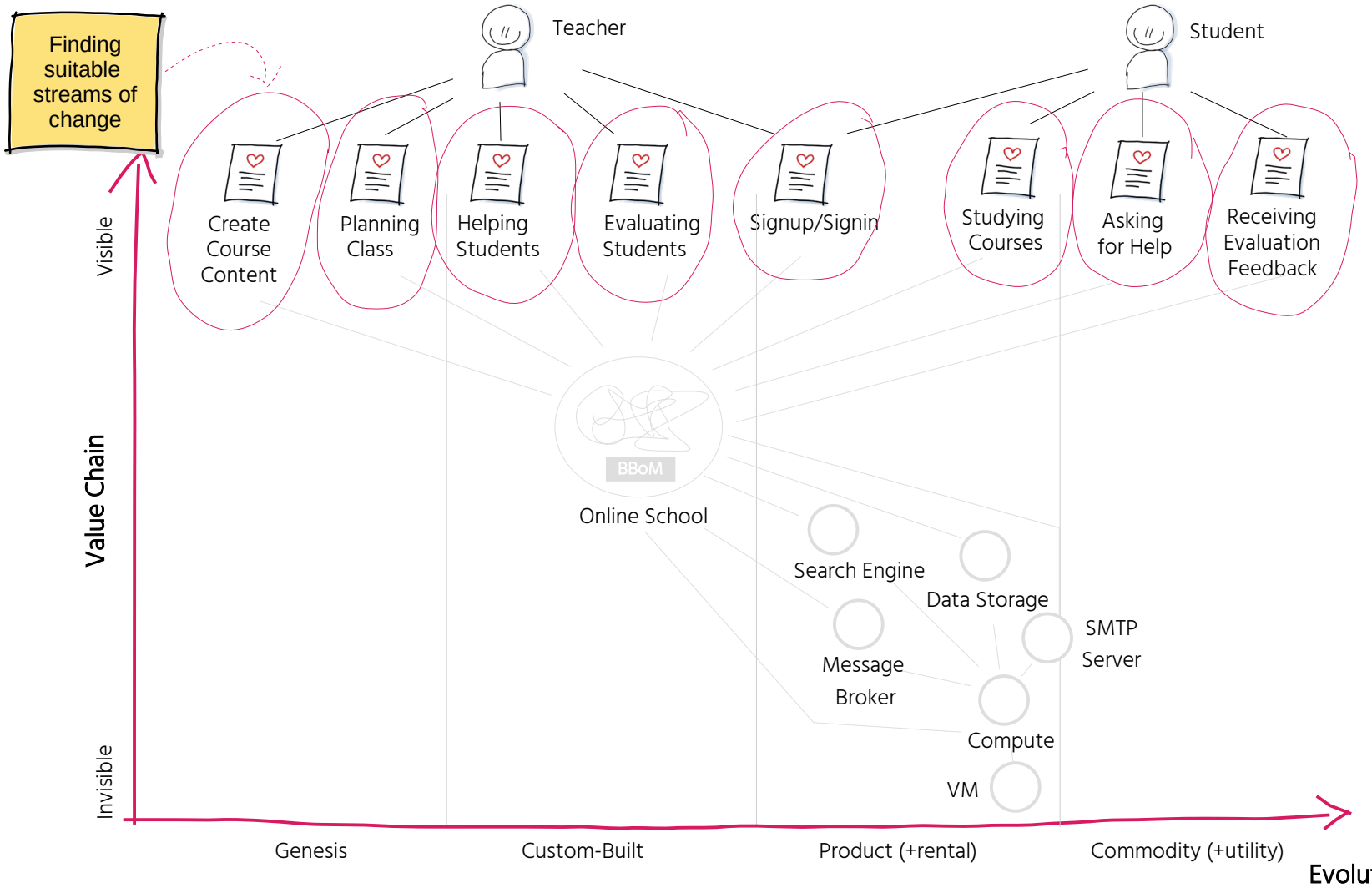
Optmizing Flow of Change



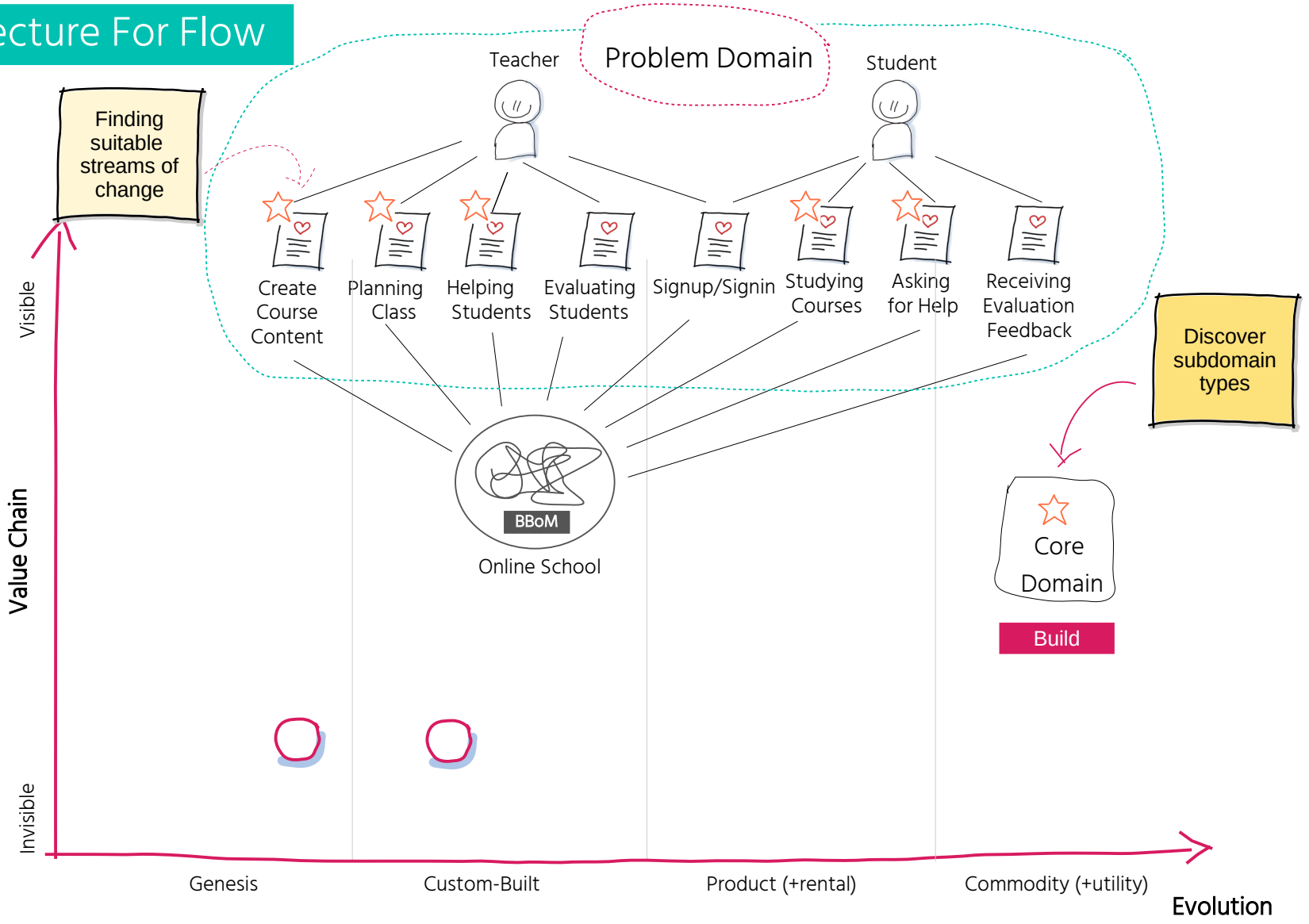
Architecture for Flow



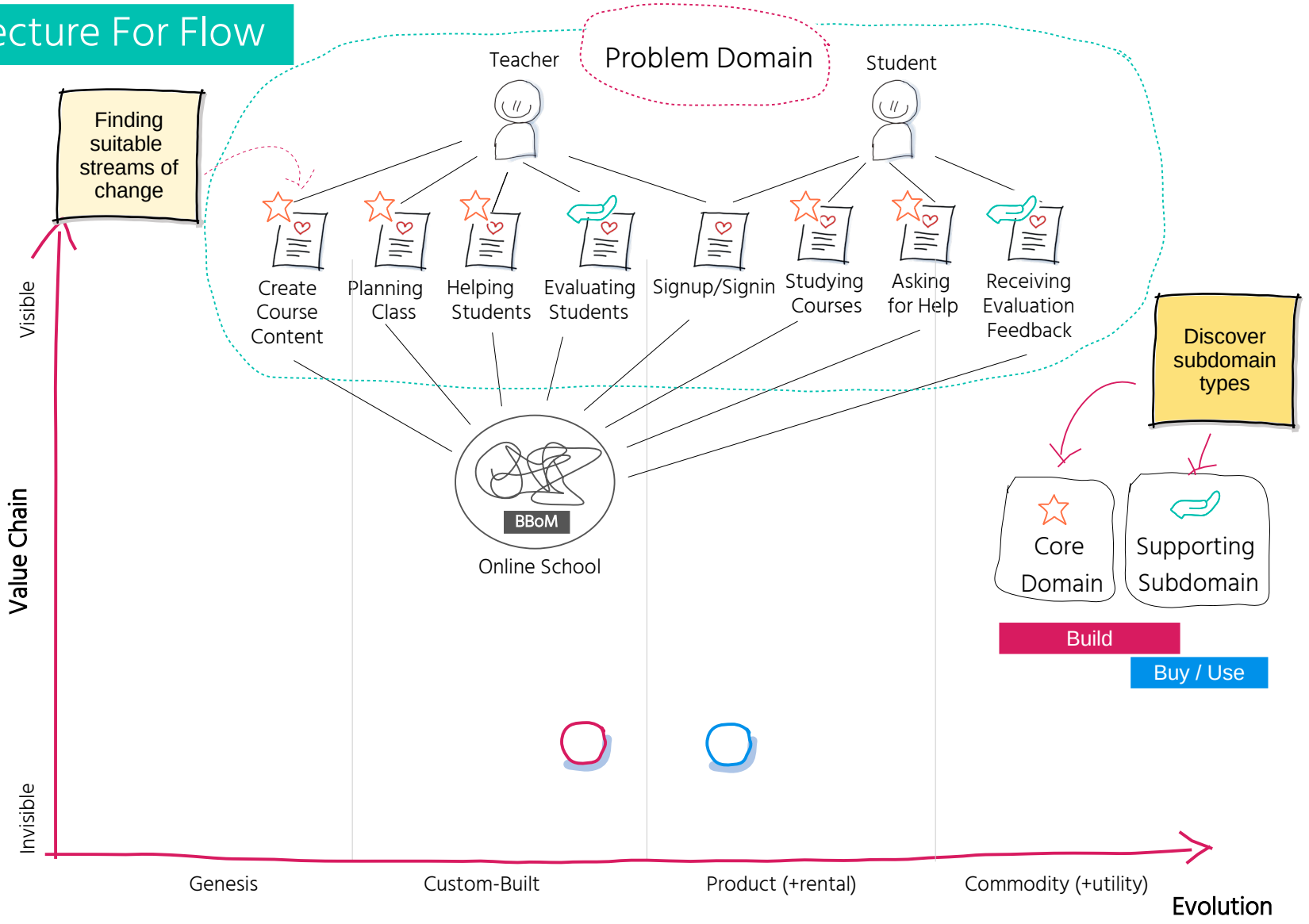
Architecture For Flow



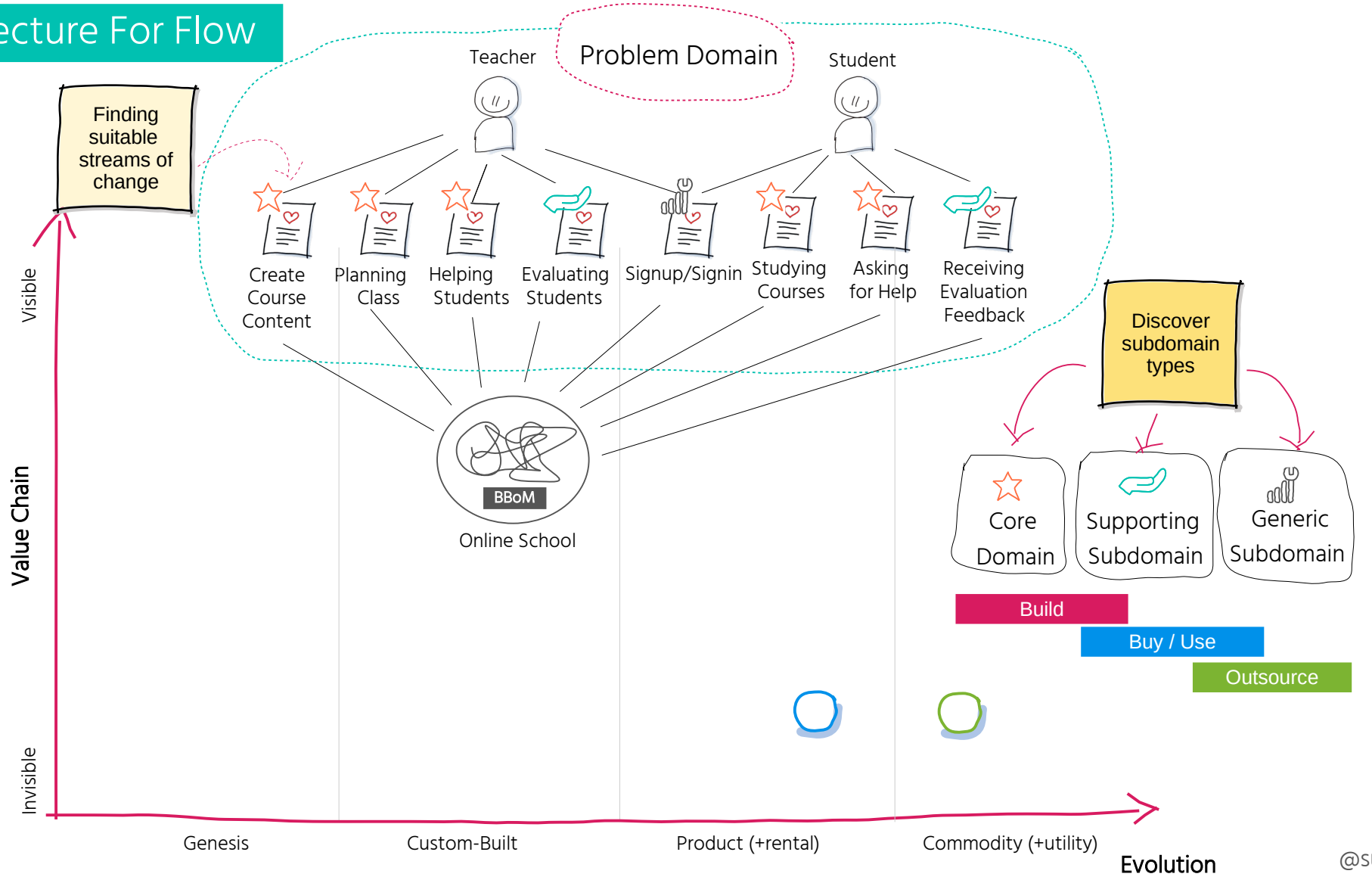
Architecture For Flow



Architecture For Flow

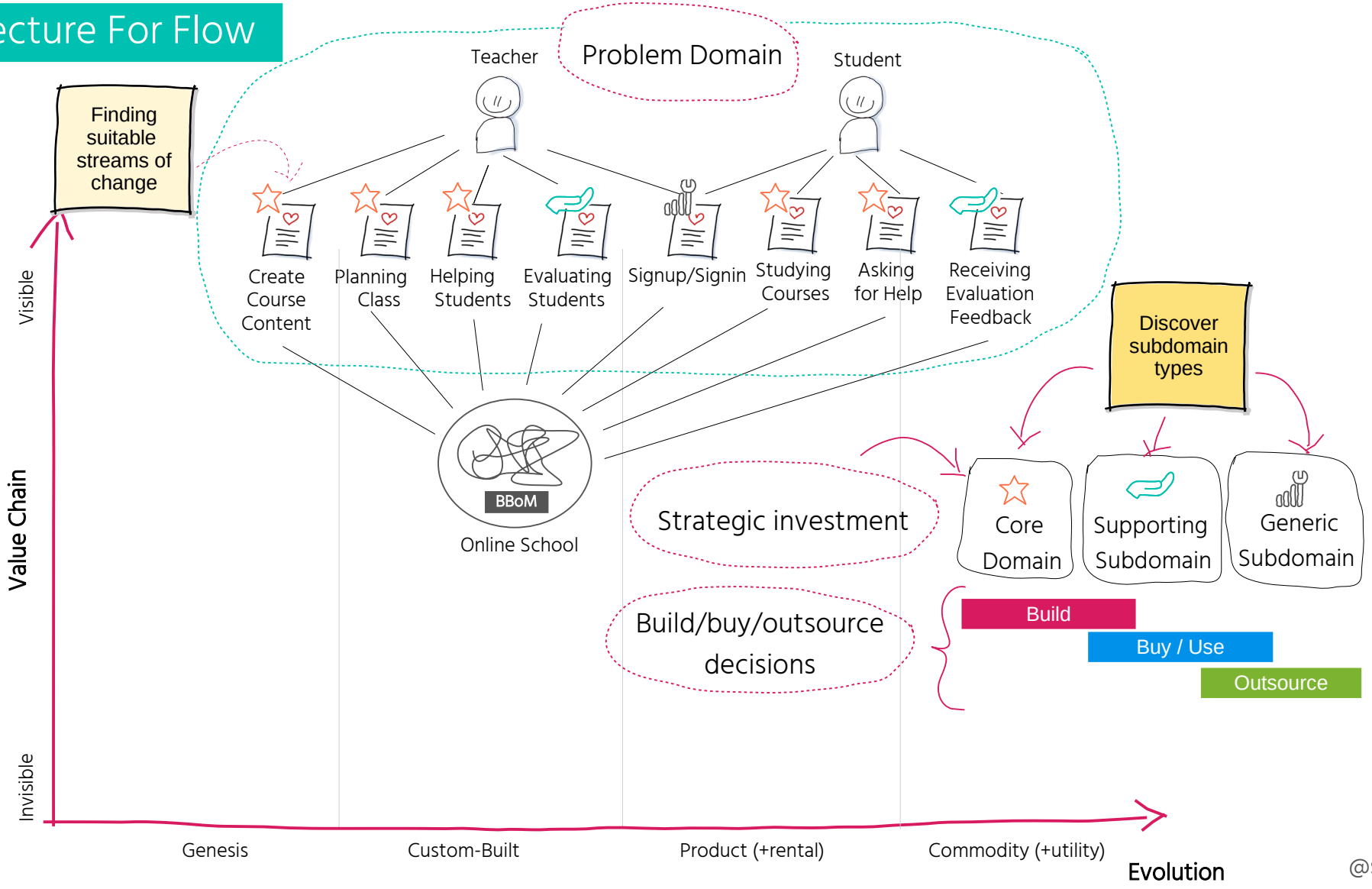


Architecture For Flow



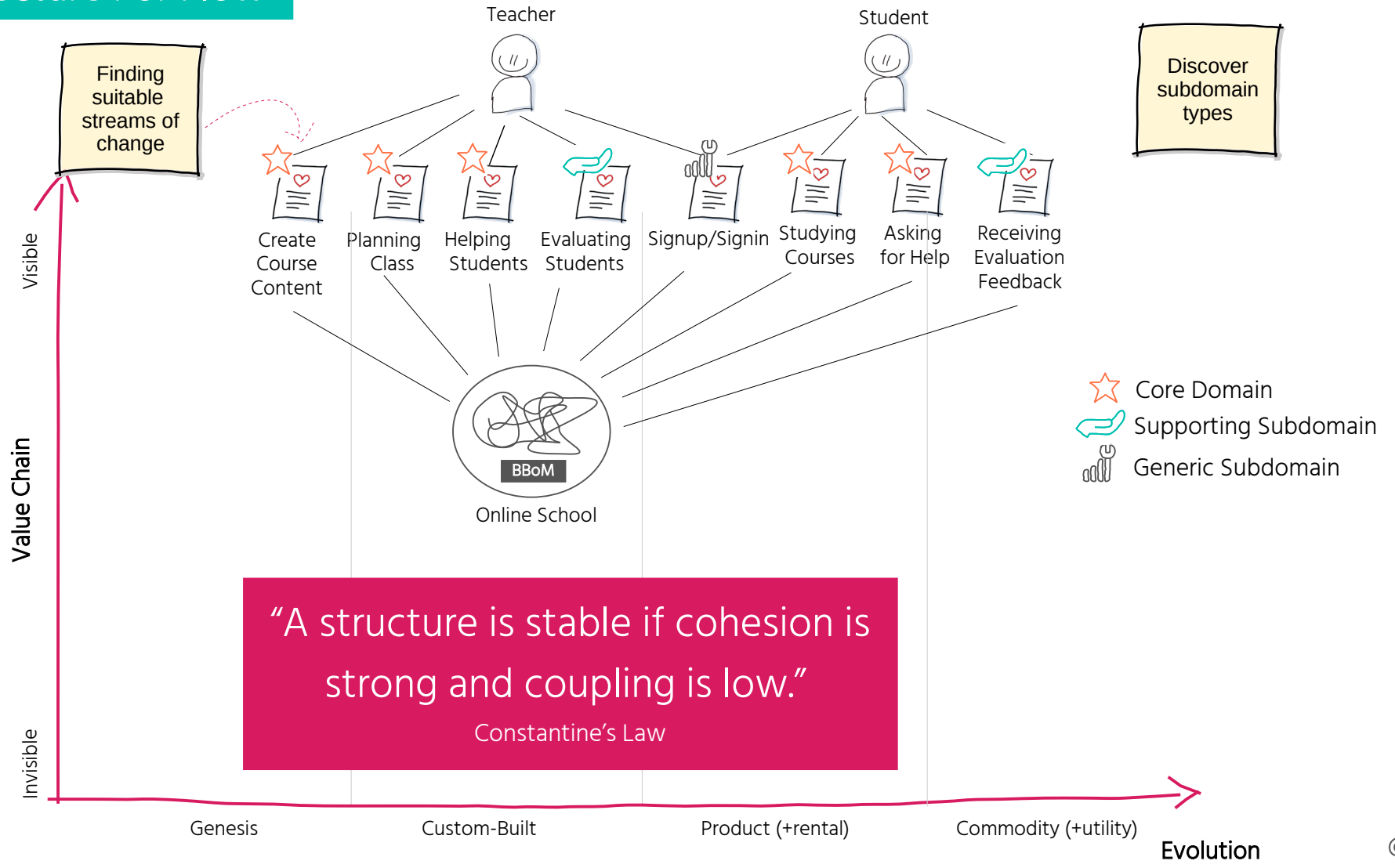
@suksr

Architecture For Flow



@suksr

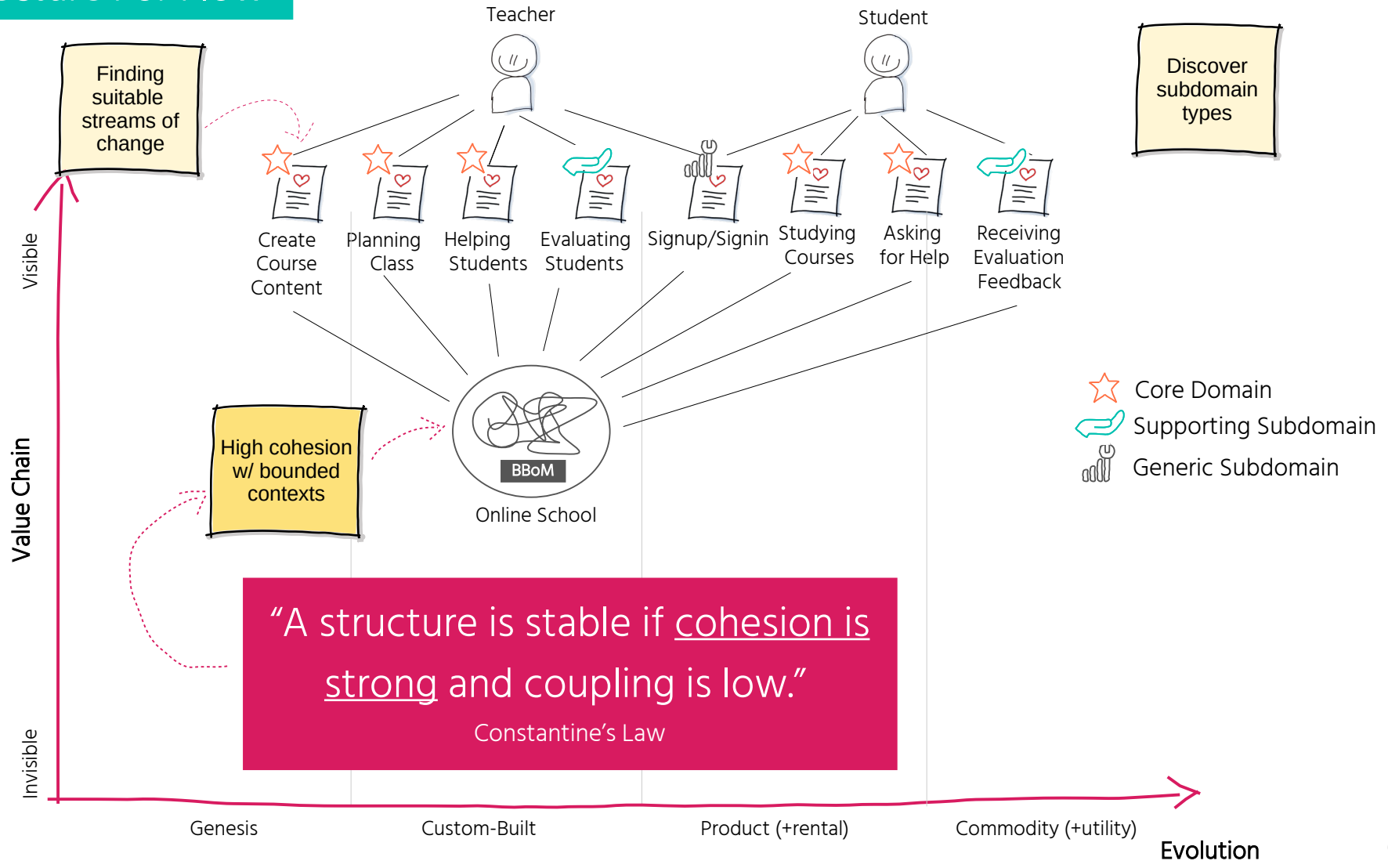
Architecture For Flow



"A structure is stable if cohesion is strong and coupling is low."

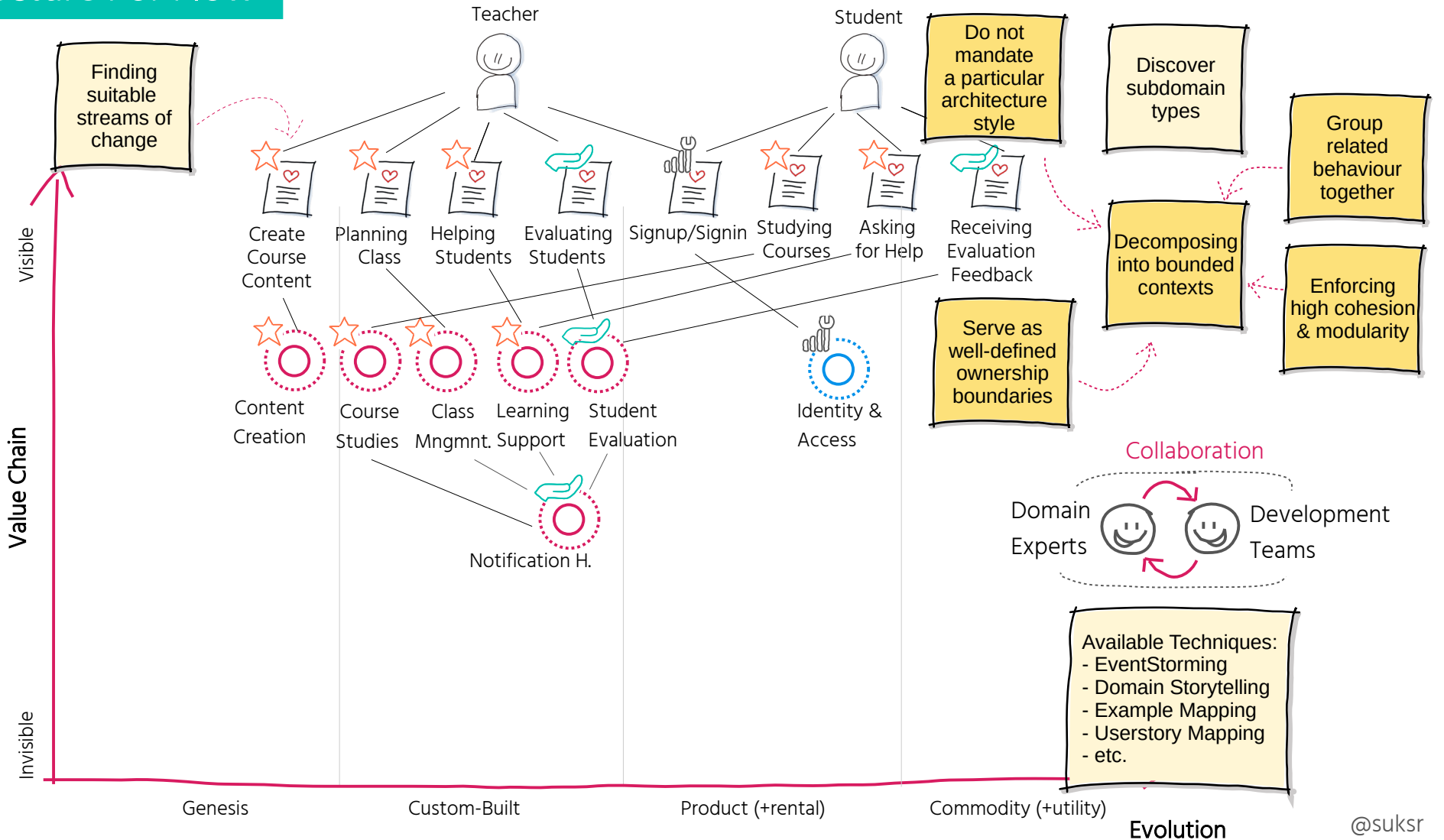
Constantine's Law

Architecture For Flow



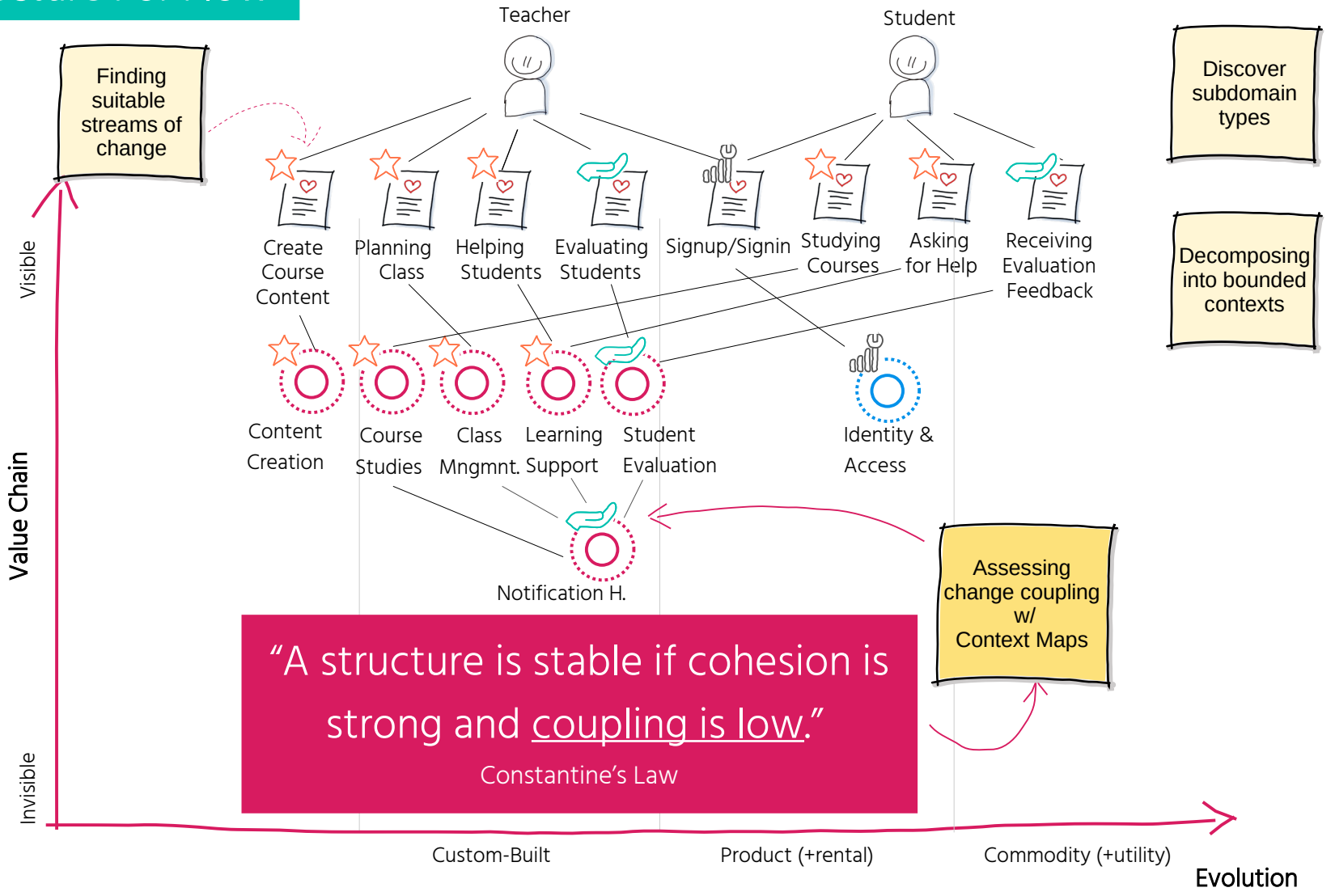
@suksr

Architecture For Flow



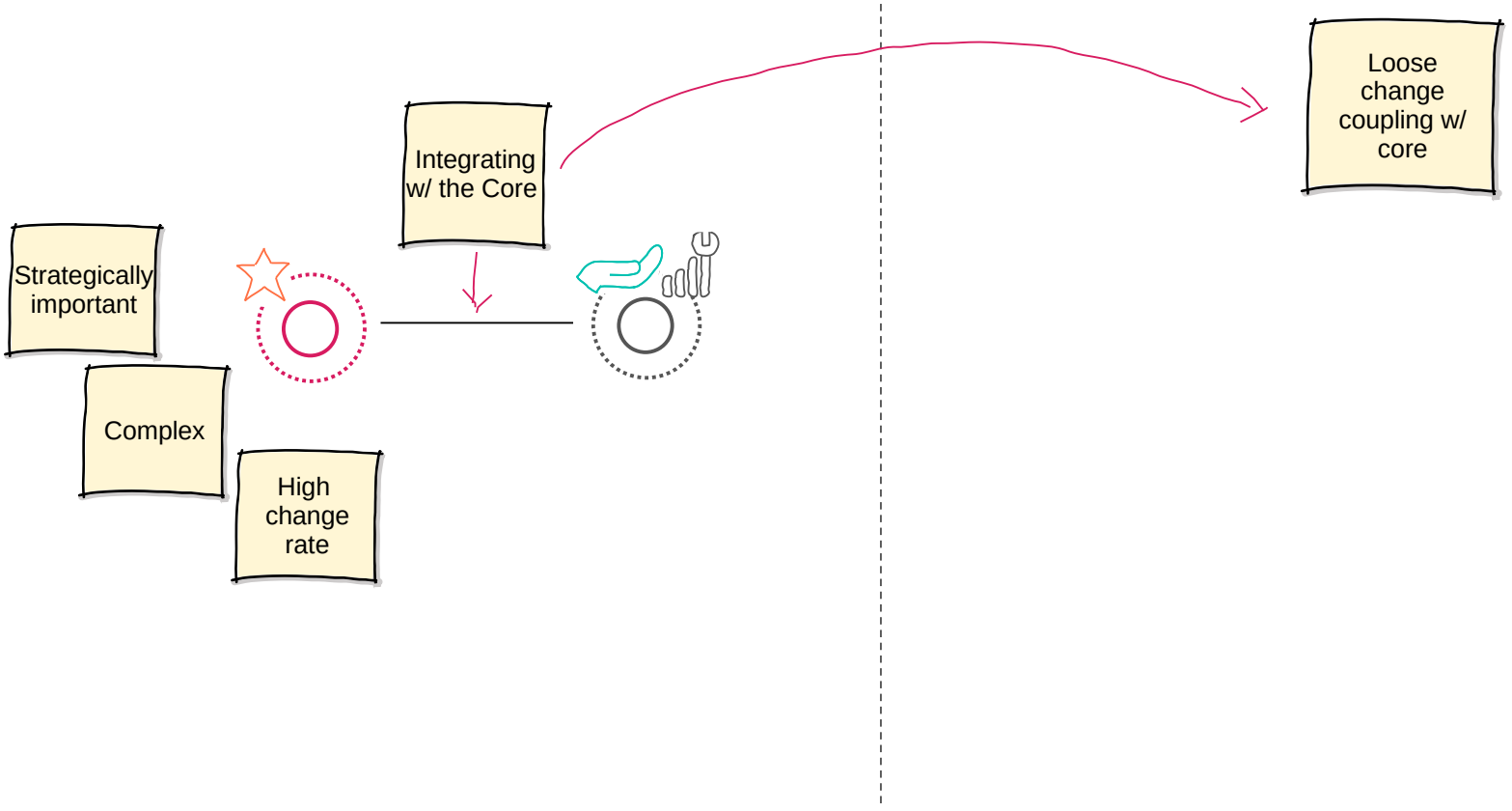
@suksr

Architecture For Flow

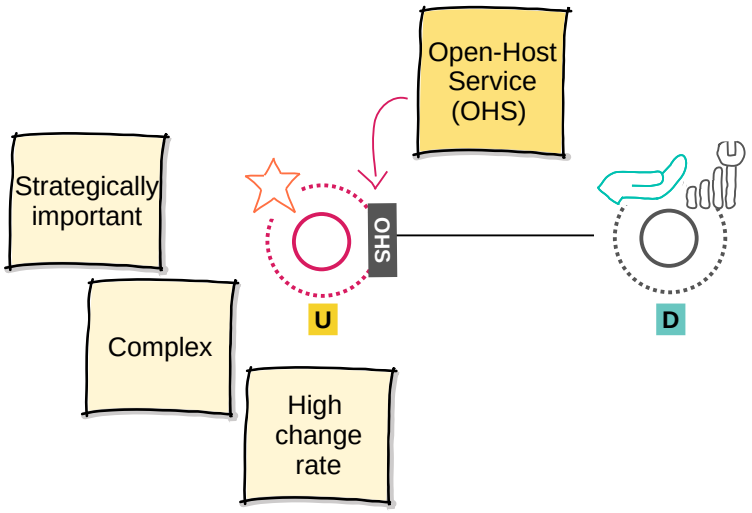


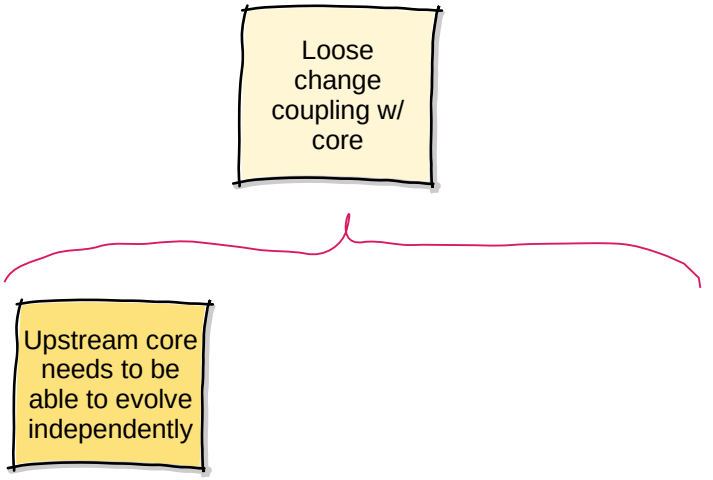
@suksr

Assessing Change Coupling w/ Context Maps

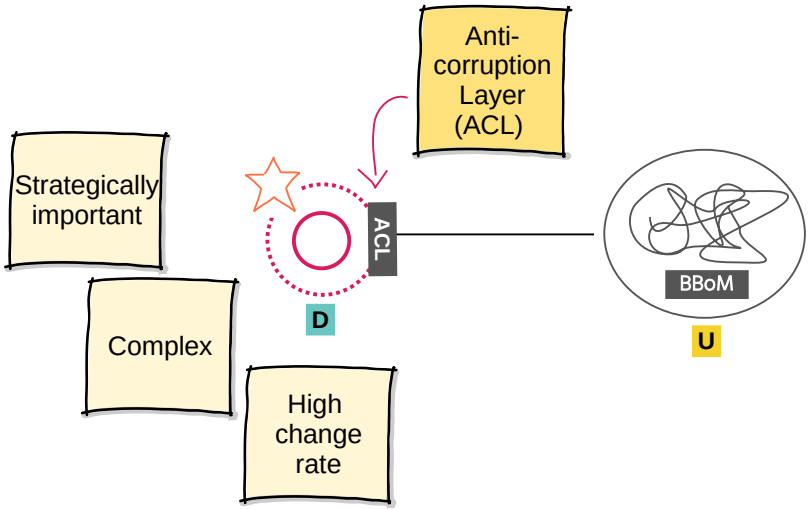


Assessing Change Coupling w/ Context Maps





Assessing Change Coupling w/ Context Maps

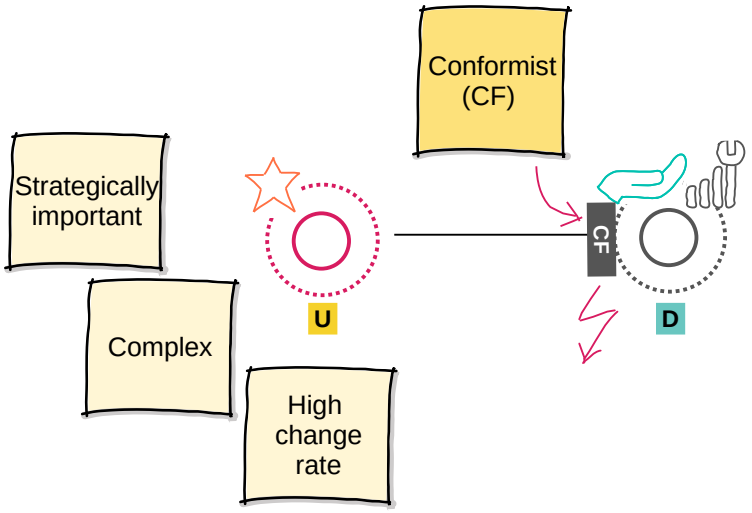


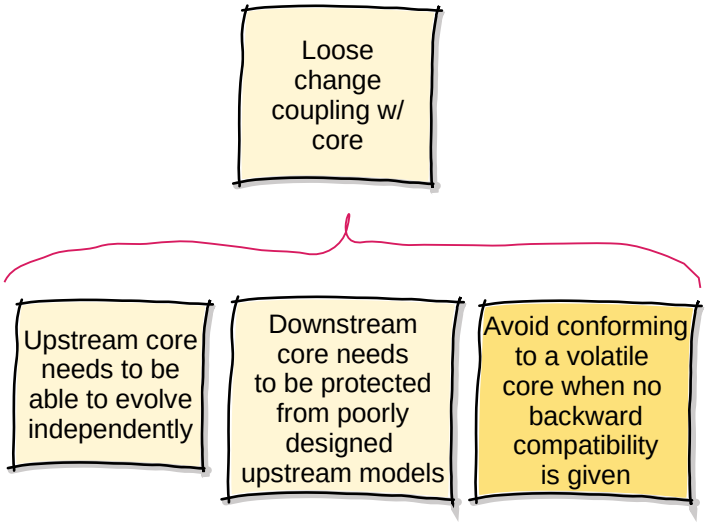
Loose change coupling w/ core

Upstream core needs to be able to evolve independently

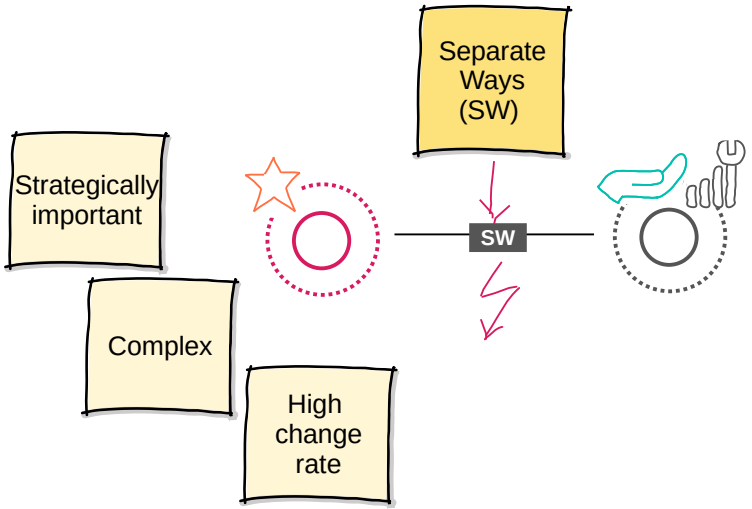
Downstream core needs to be protected from poorly designed upstream models

Assessing Change Coupling w/ Context Maps





Assessing Change Coupling w/ Context Maps



Loose change coupling w/ core

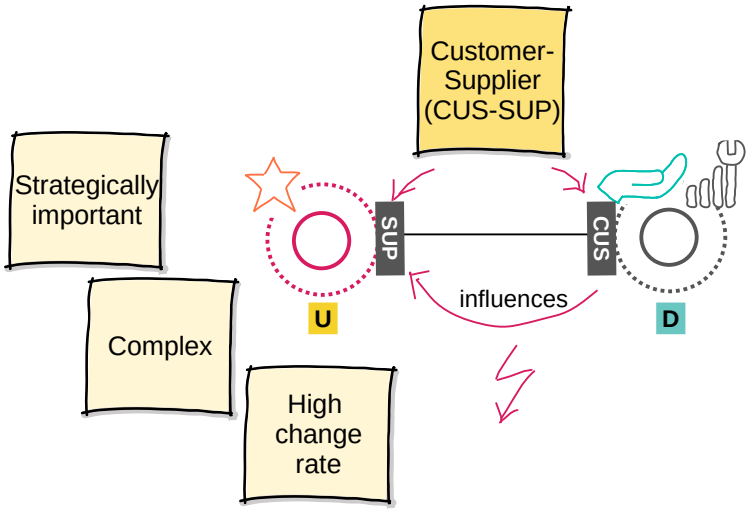
Upstream core needs to be able to evolve independently

Downstream core needs to be protected from poorly designed upstream models

Avoid conforming to a volatile core when no backward compatibility is given

Minimize duplication of complex core behaviour

Assessing Change Coupling w/ Context Maps



Loose change coupling w/ core

Upstream core needs to be able to evolve independently

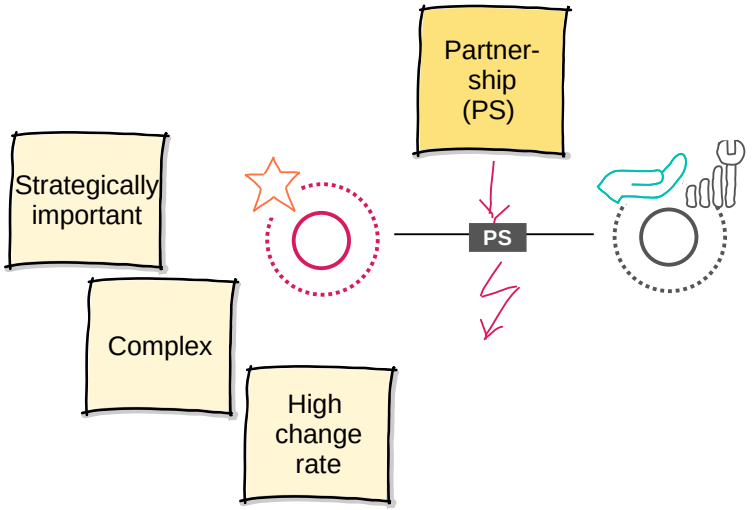
Downstream core needs to be protected from poorly designed upstream models

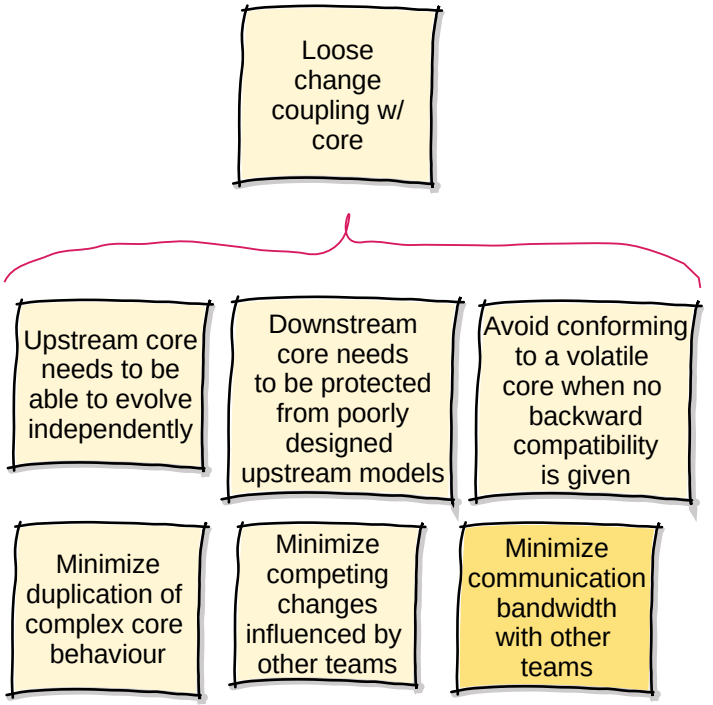
Avoid conforming to a volatile core when no backward compatibility is given

Minimize duplication of complex core behaviour

Minimize competing changes influenced by other teams

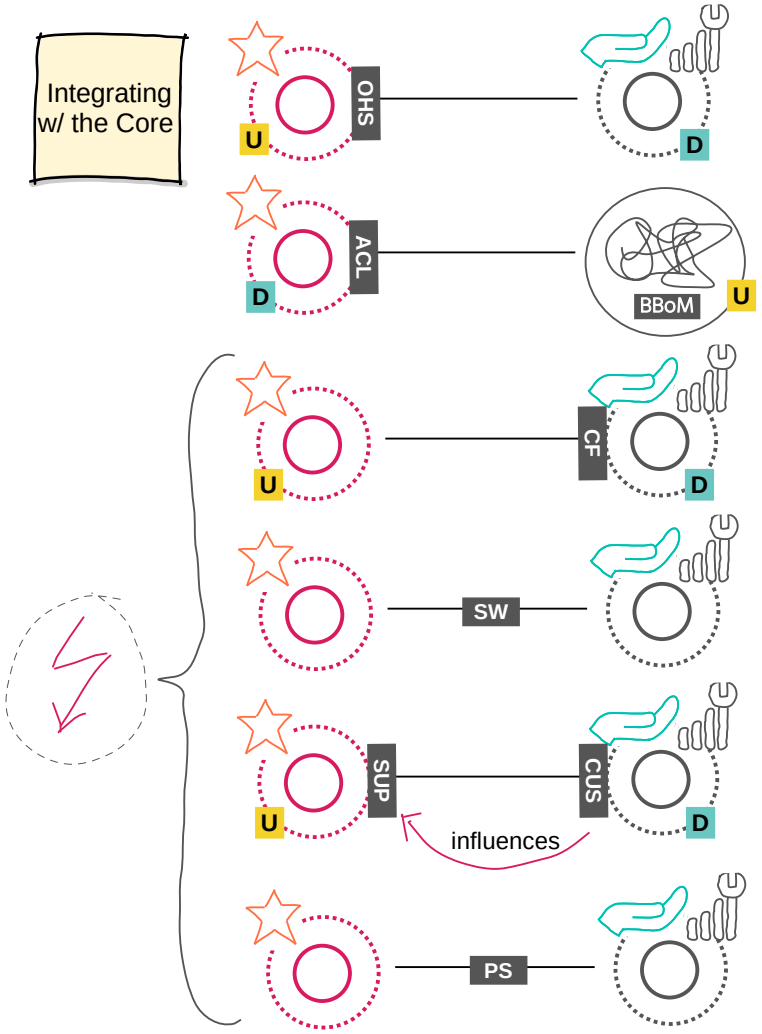
Assessing Change Coupling w/ Context Maps





Assessing Change Coupling w/ Context Maps

Integrating w/ the Core

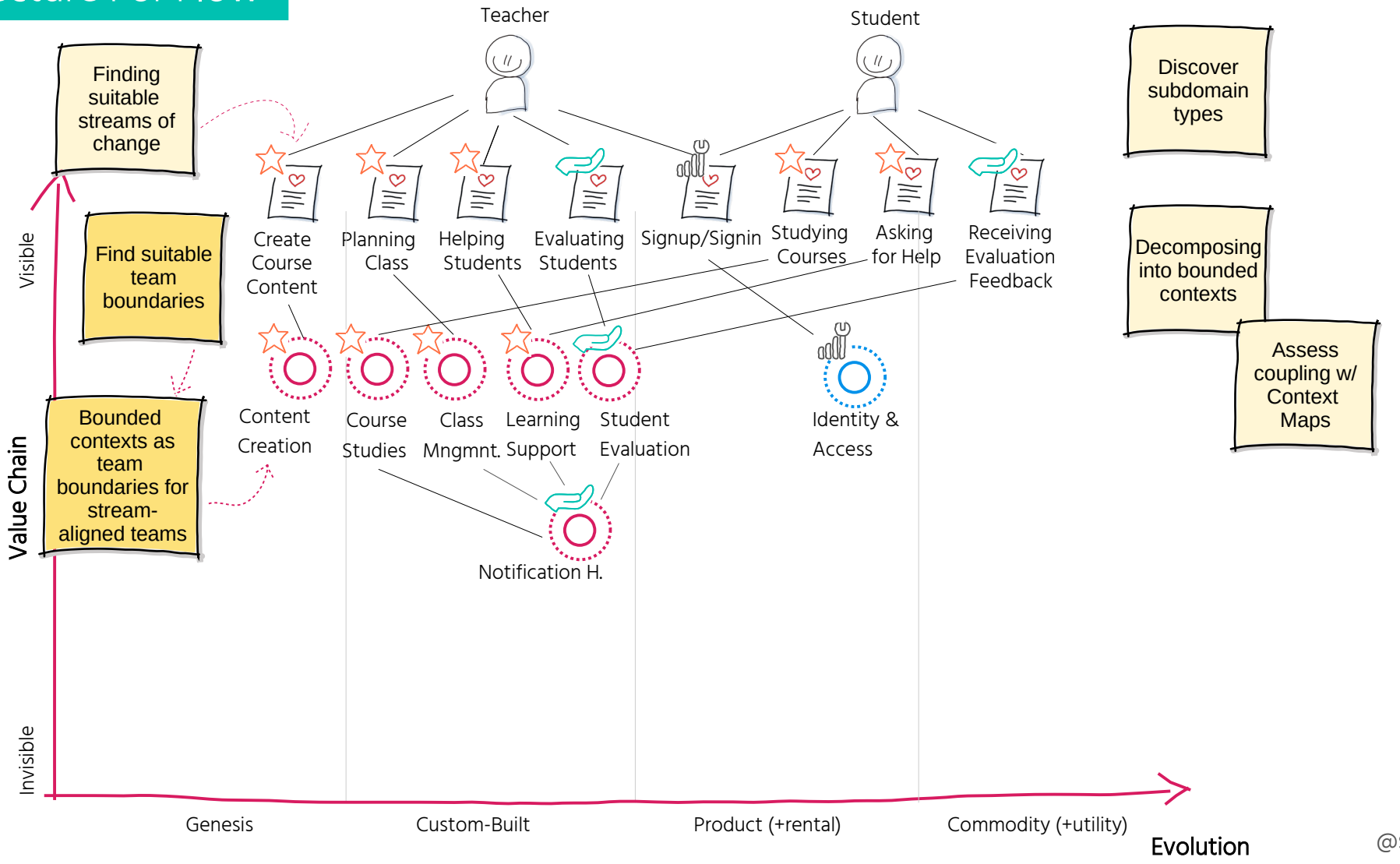


Context Maps make change coupling visible

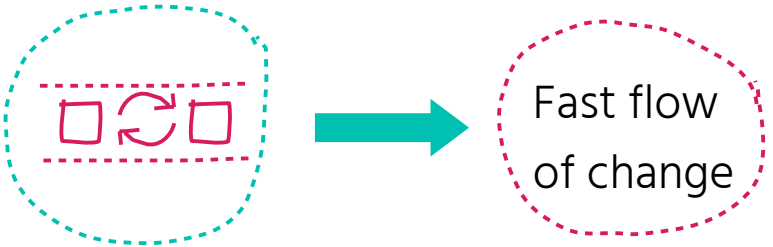
Loose change coupling w/ core

- Upstream core needs to be able to evolve independently
- Downstream core needs to be protected from poorly designed upstream models
- Avoid conforming to a volatile core when no backward compatibility is given
- Minimize duplication of complex core behaviour
- Minimize competing changes influenced by other teams
- Minimize communication bandwidth with other teams

Architecture For Flow

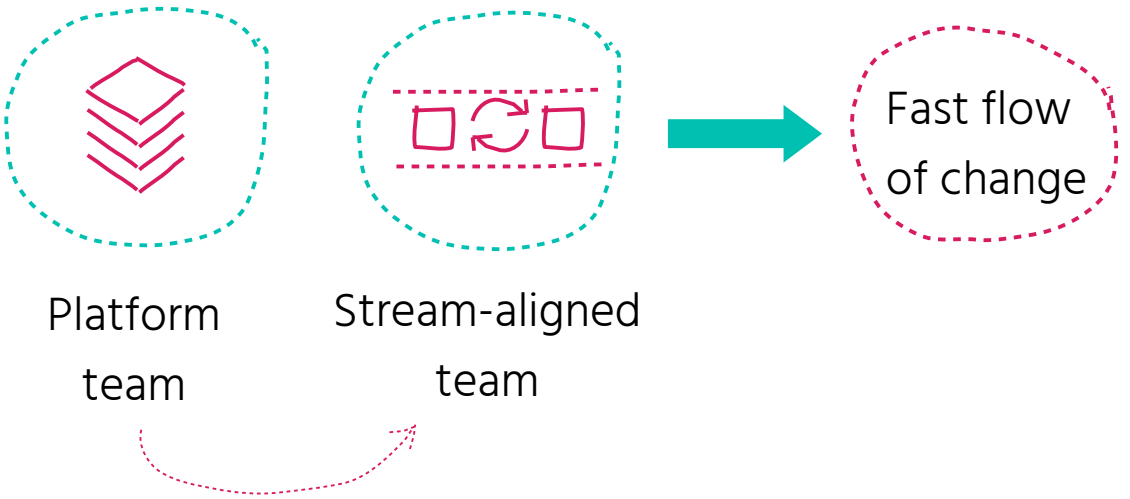


Four Team Types of Team Topologies

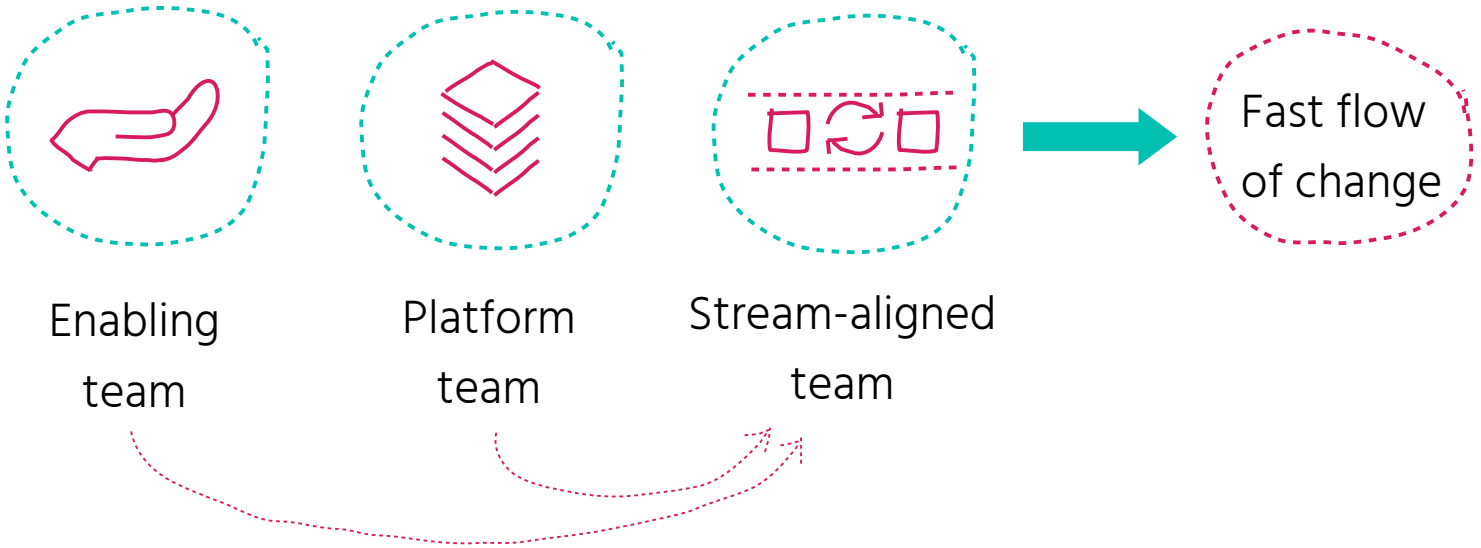


Stream-aligned
team

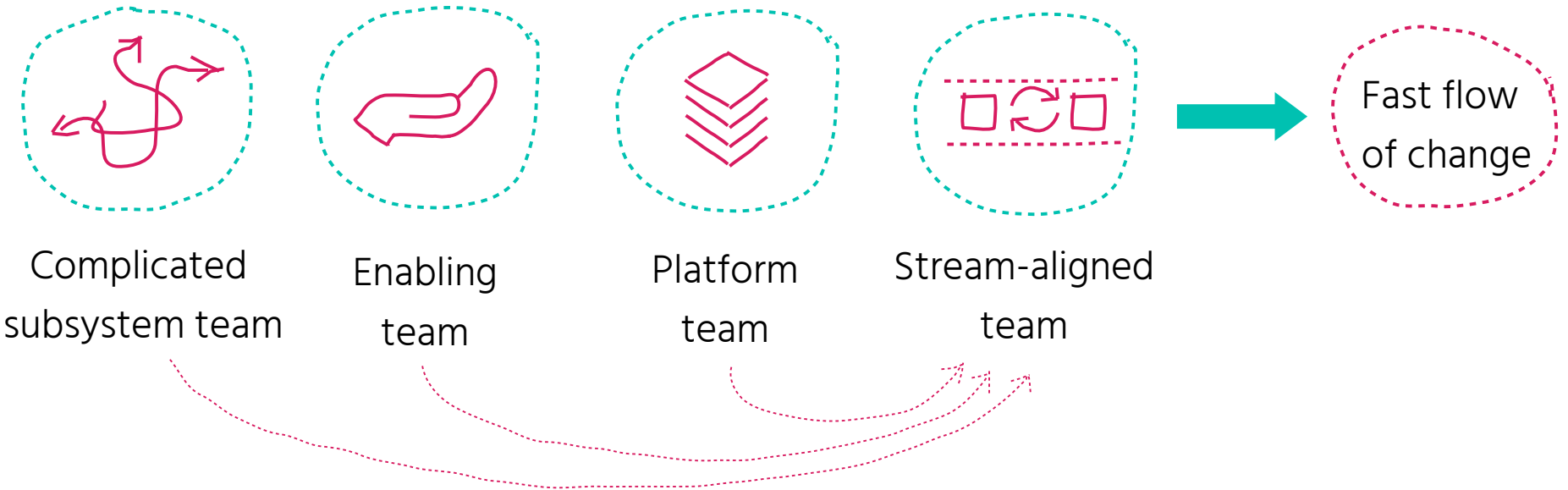
Four Team Types of Team Topologies



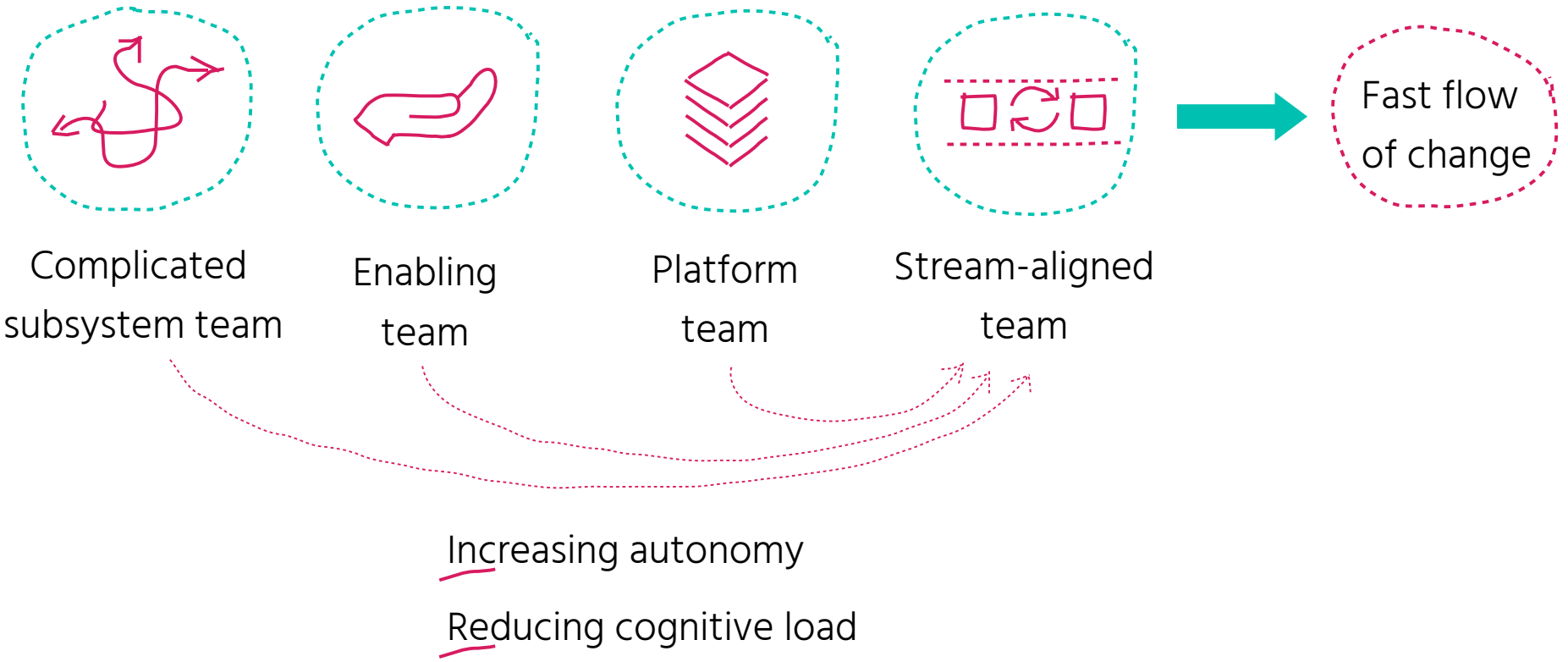
Four Team Types of Team Topologies



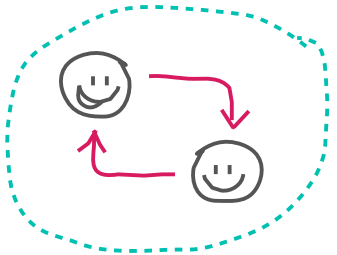
Four Team Types of Team Topologies



Four Team Types of Team Topologies

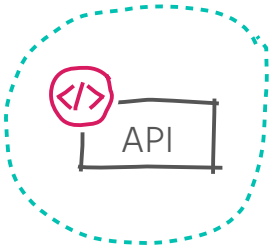


Three Interaction Modes



Collaboration

Rapid discovery



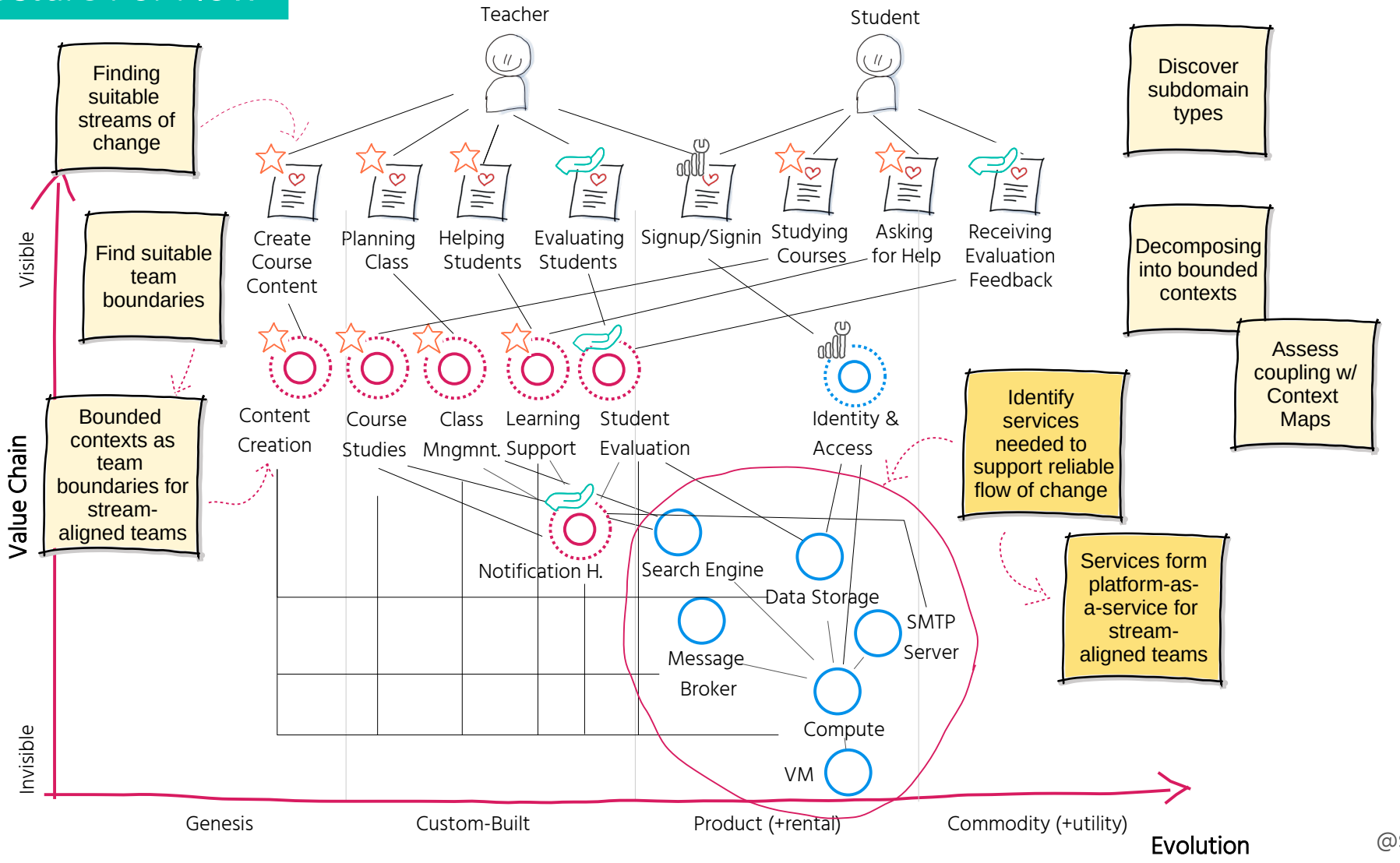
X-as-a-Service

Predictable delivery

Facilitating

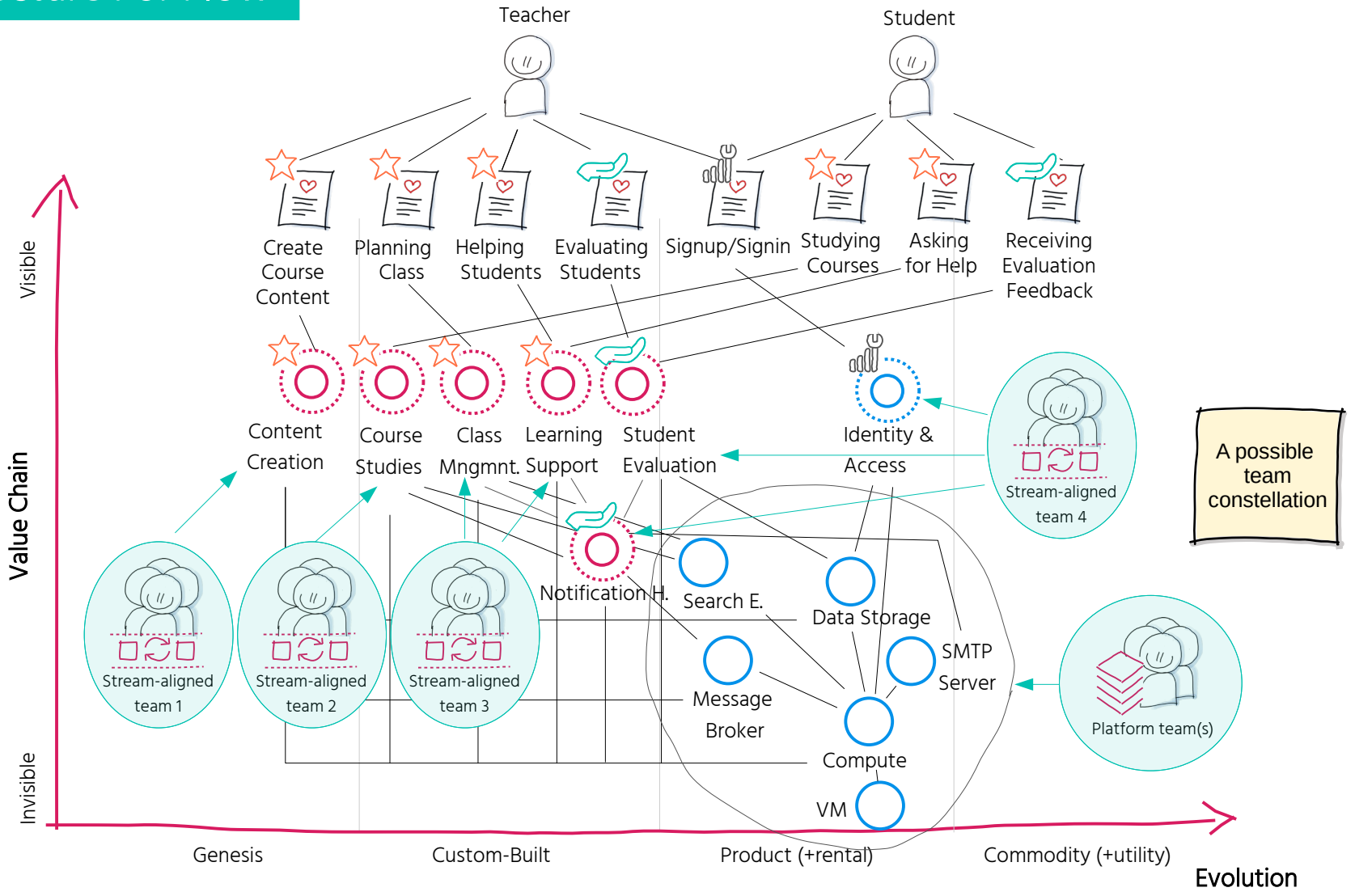
Active help

Architecture For Flow



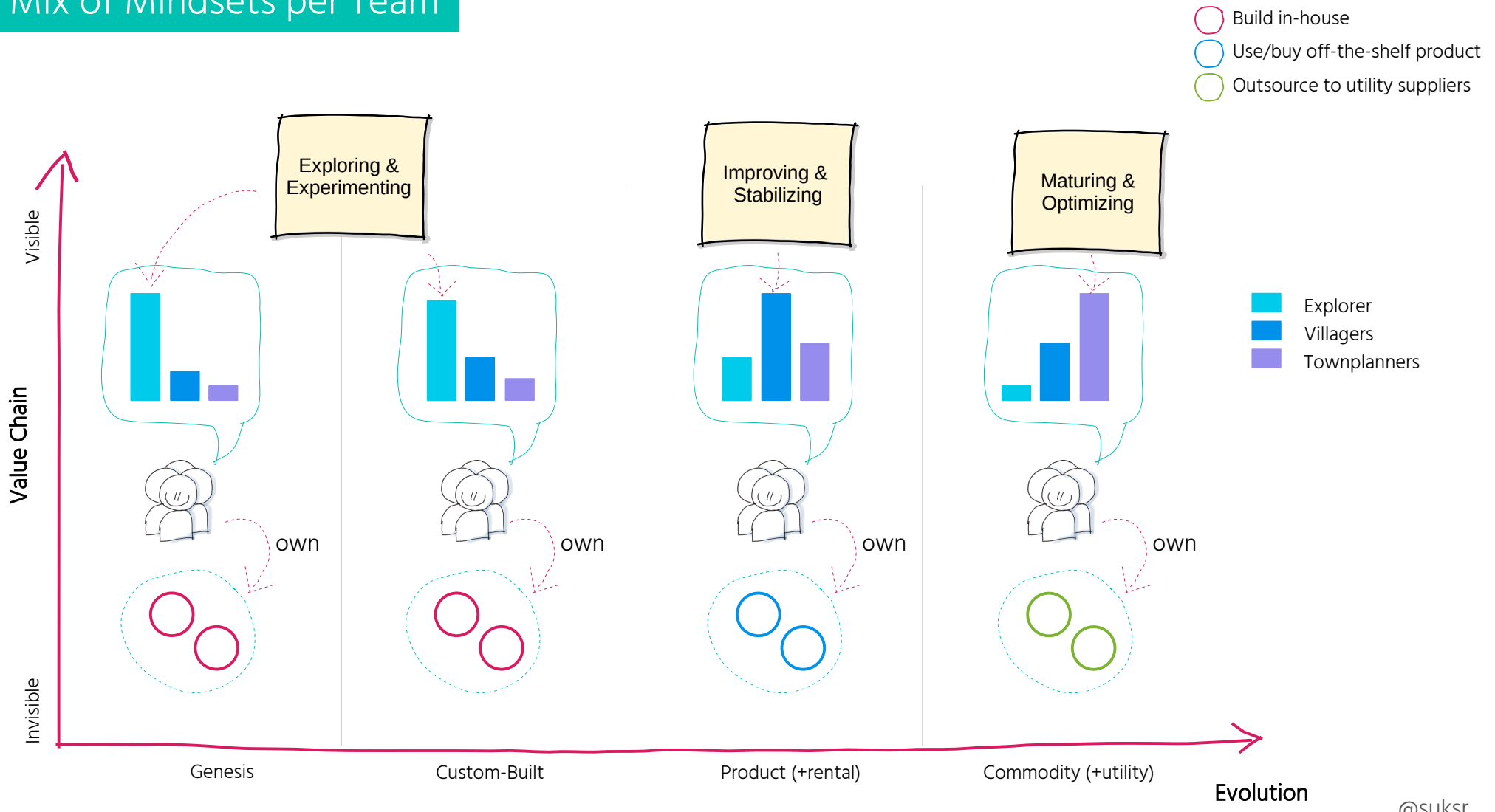
@suksr

Architecture For Flow



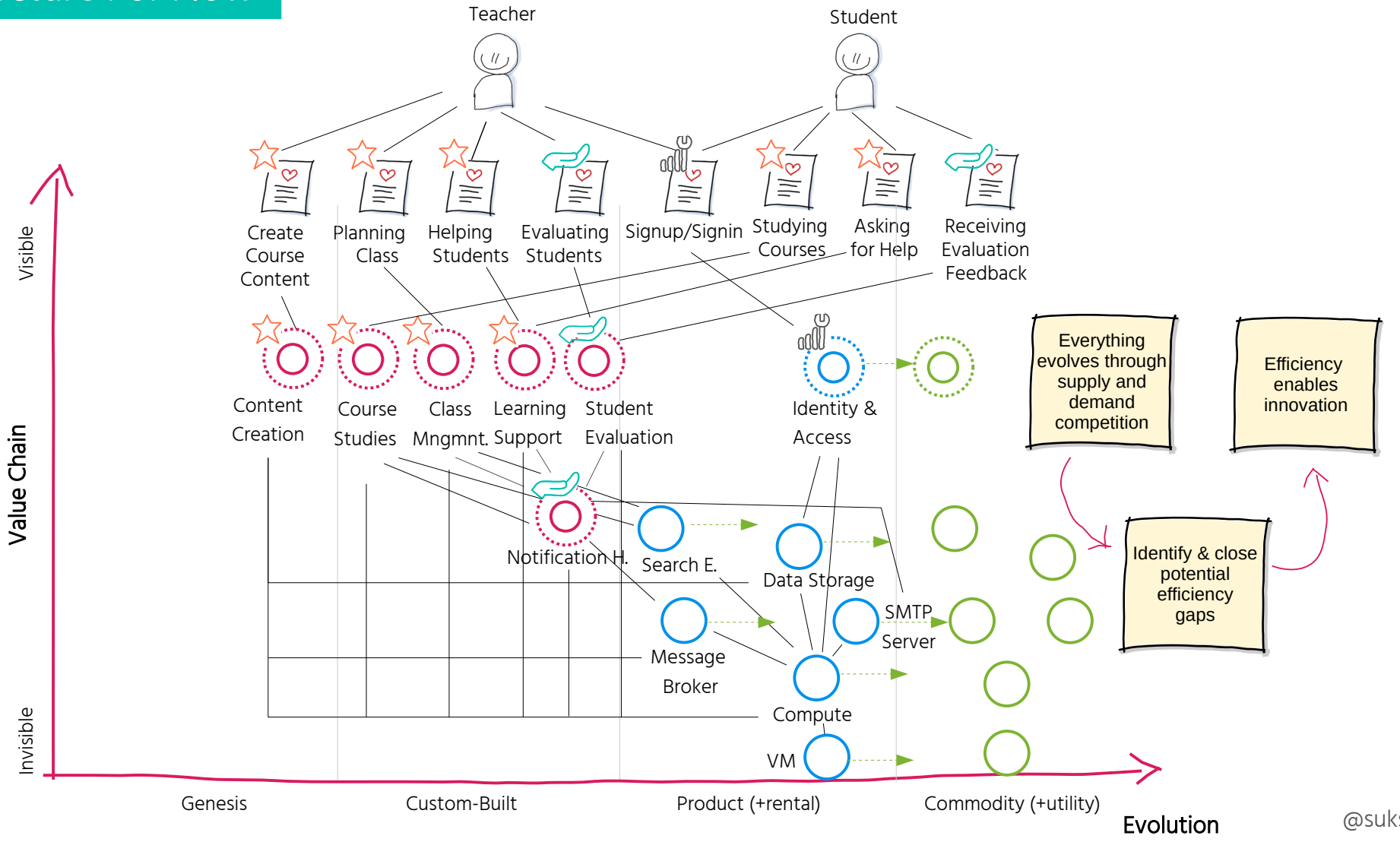
@suksr

A Mix of Mindsets per Team



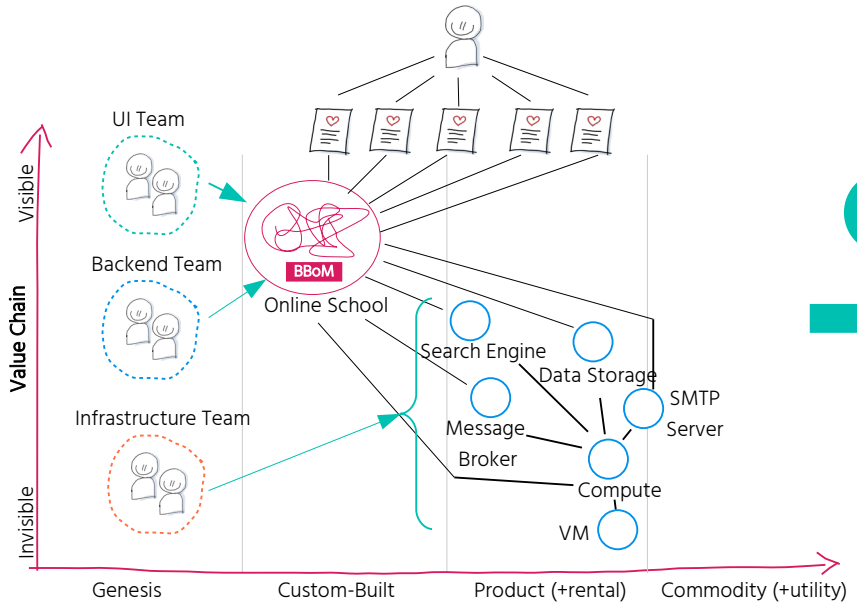
@suksr

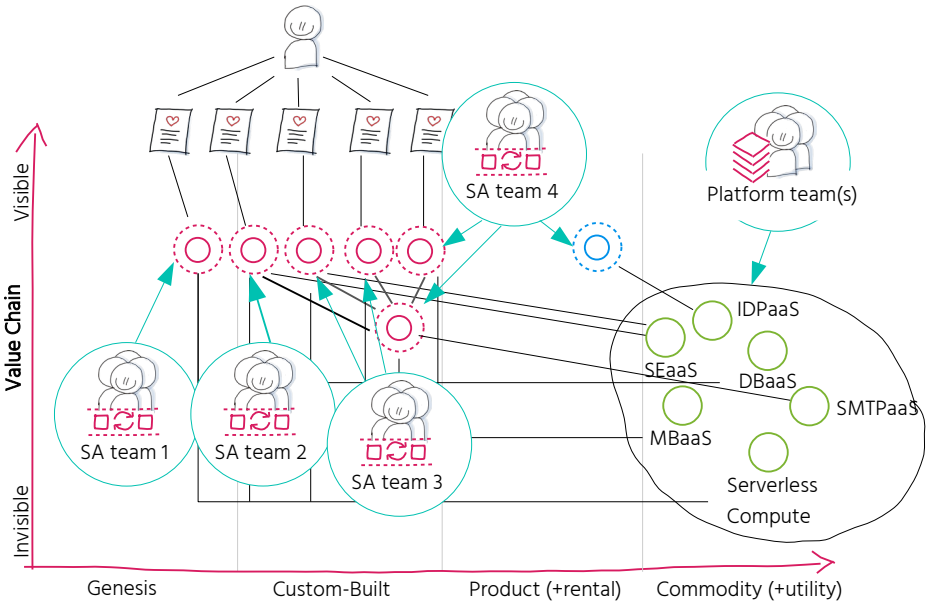
Architecture For Flow



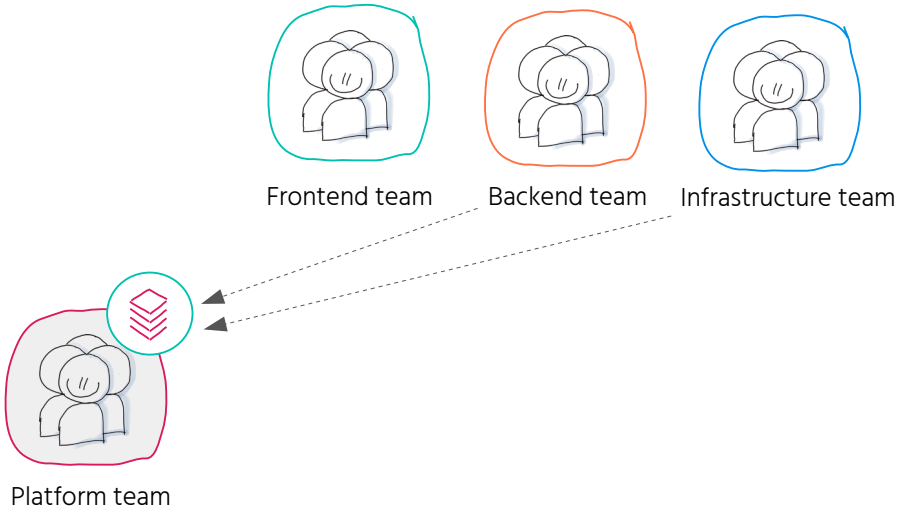
@suksr

How to transition?

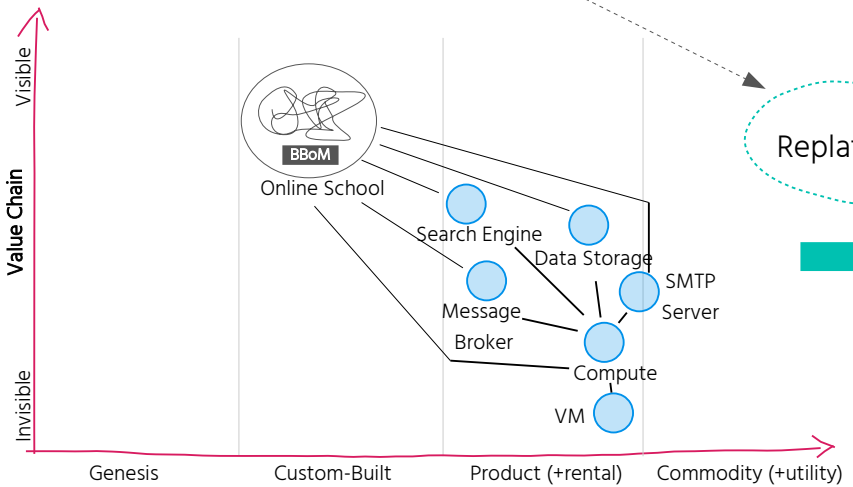


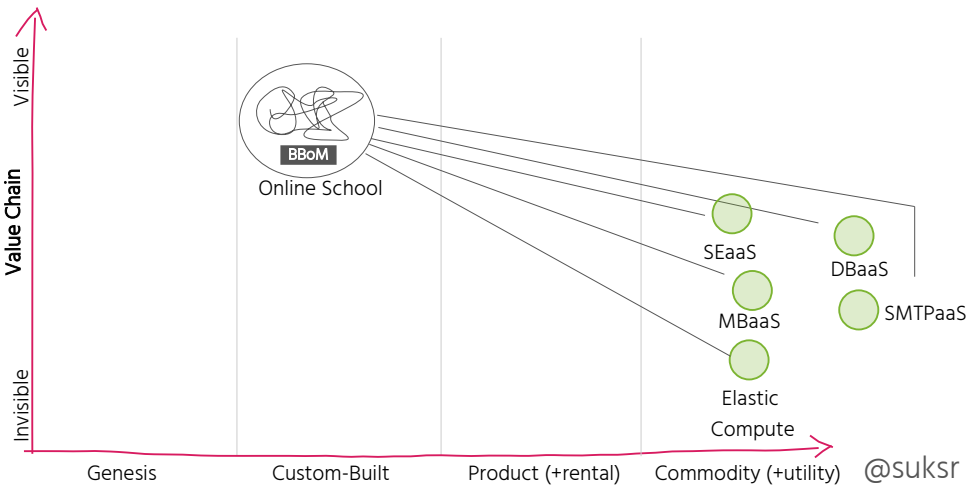


Evolution of Team Topologies



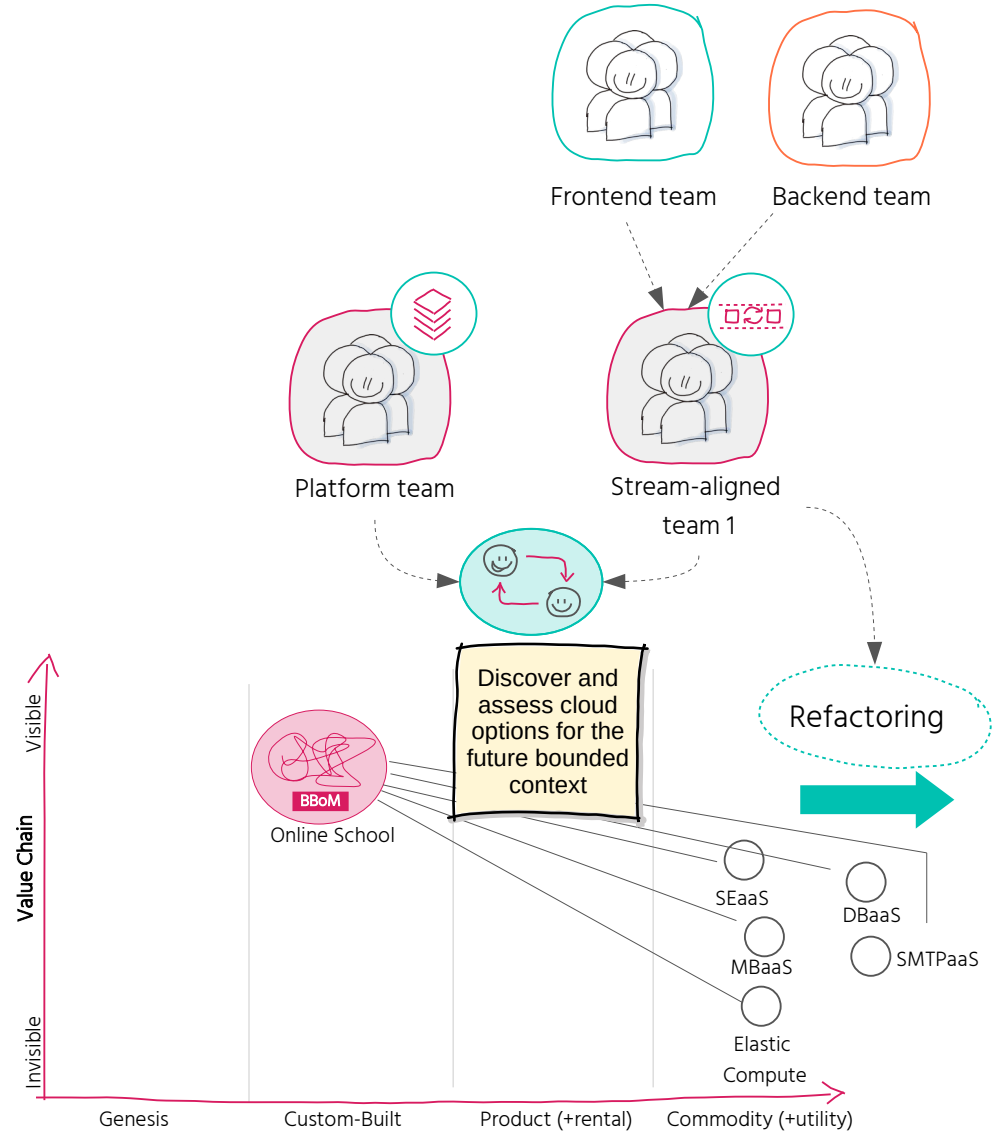
Replatforming

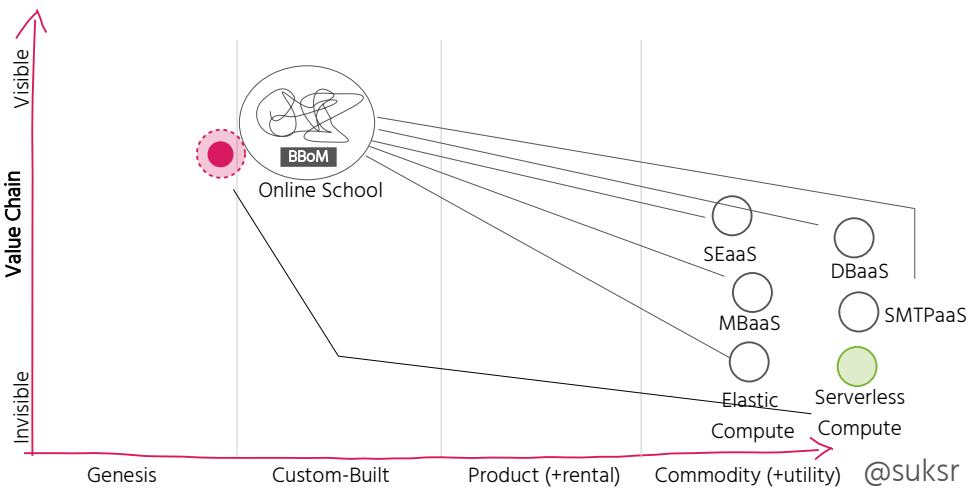




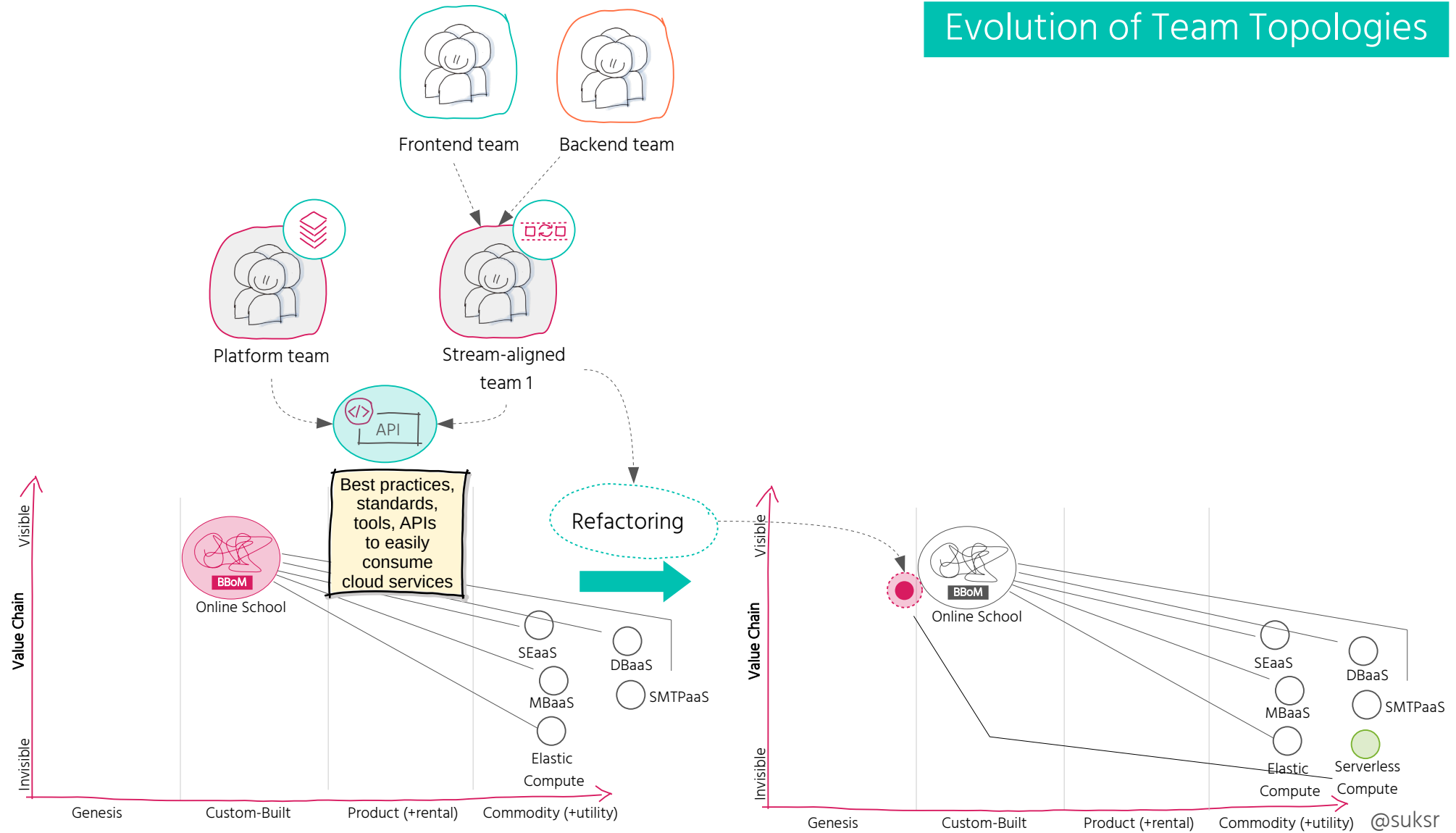
@suksr

Evolution of Team Topologies

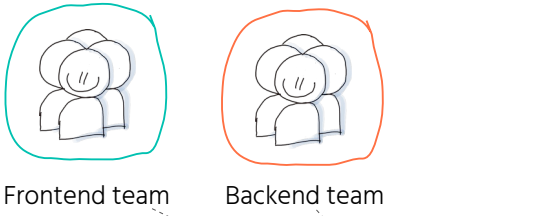


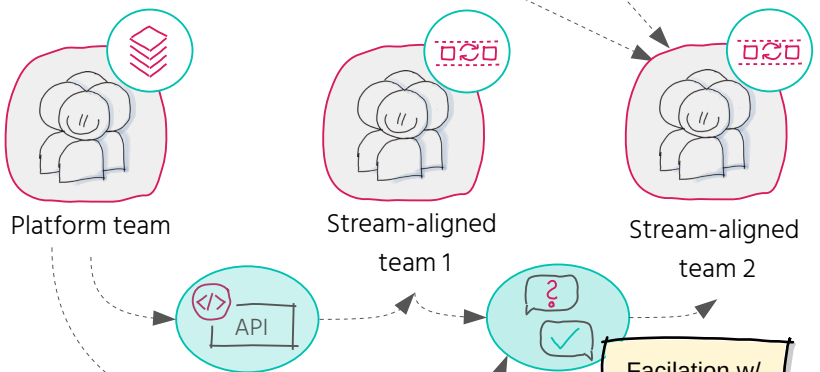


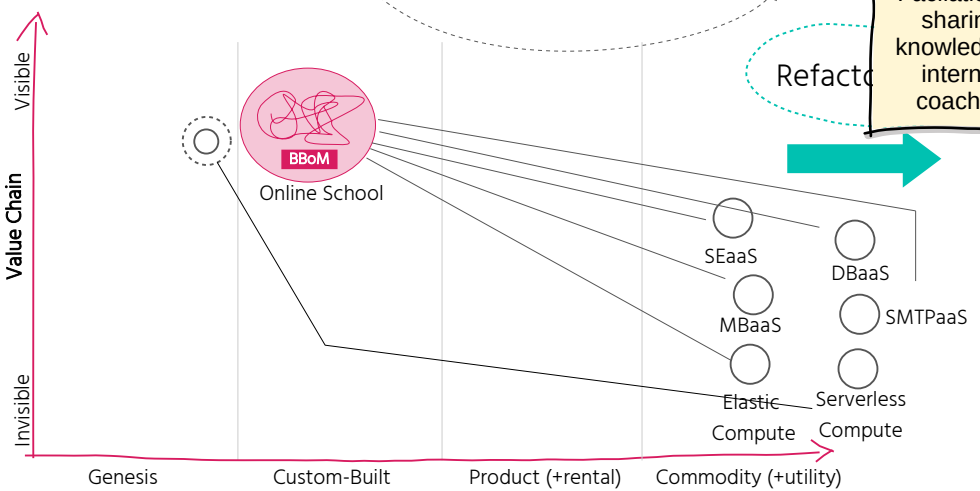
Evolution of Team Topologies

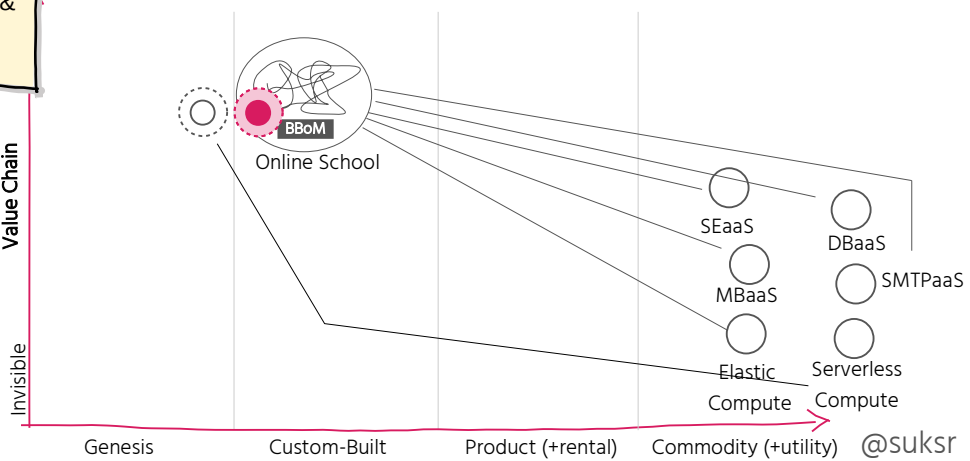


Evolution of Team Topologies

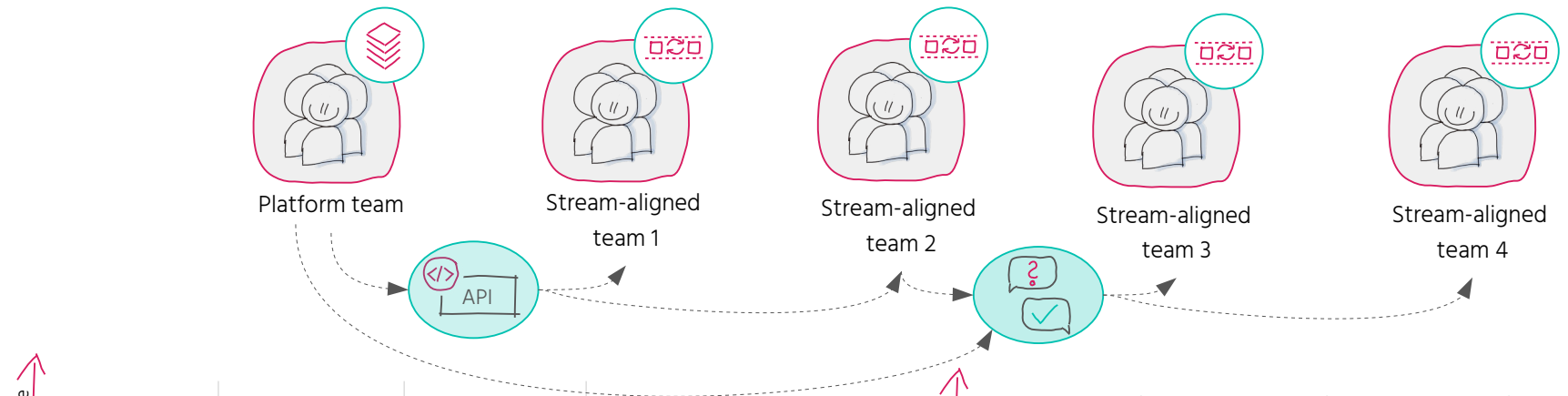


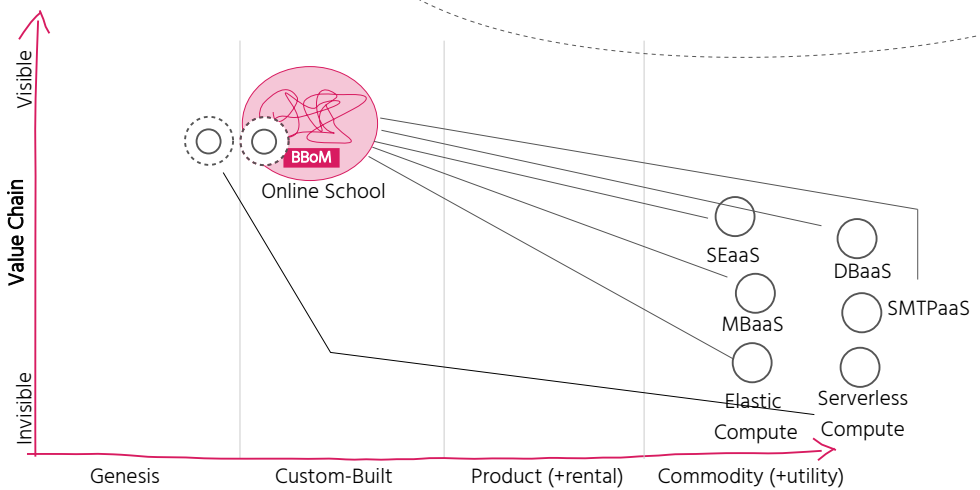


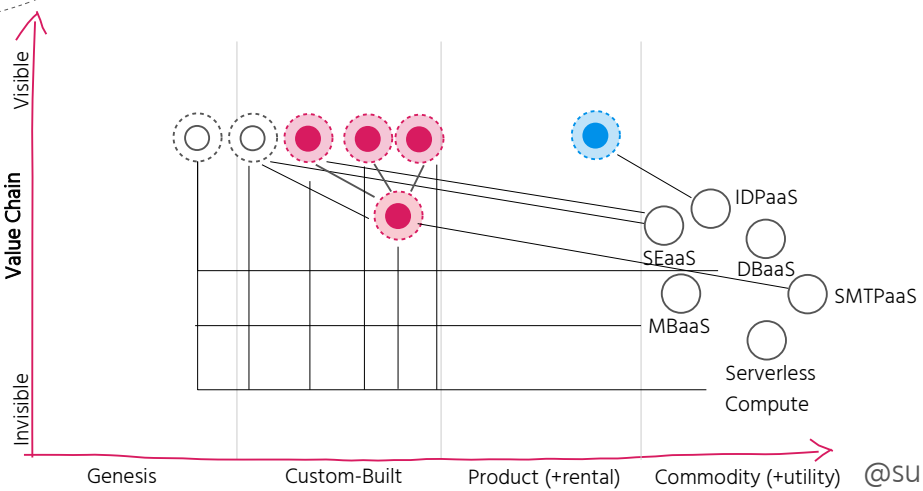




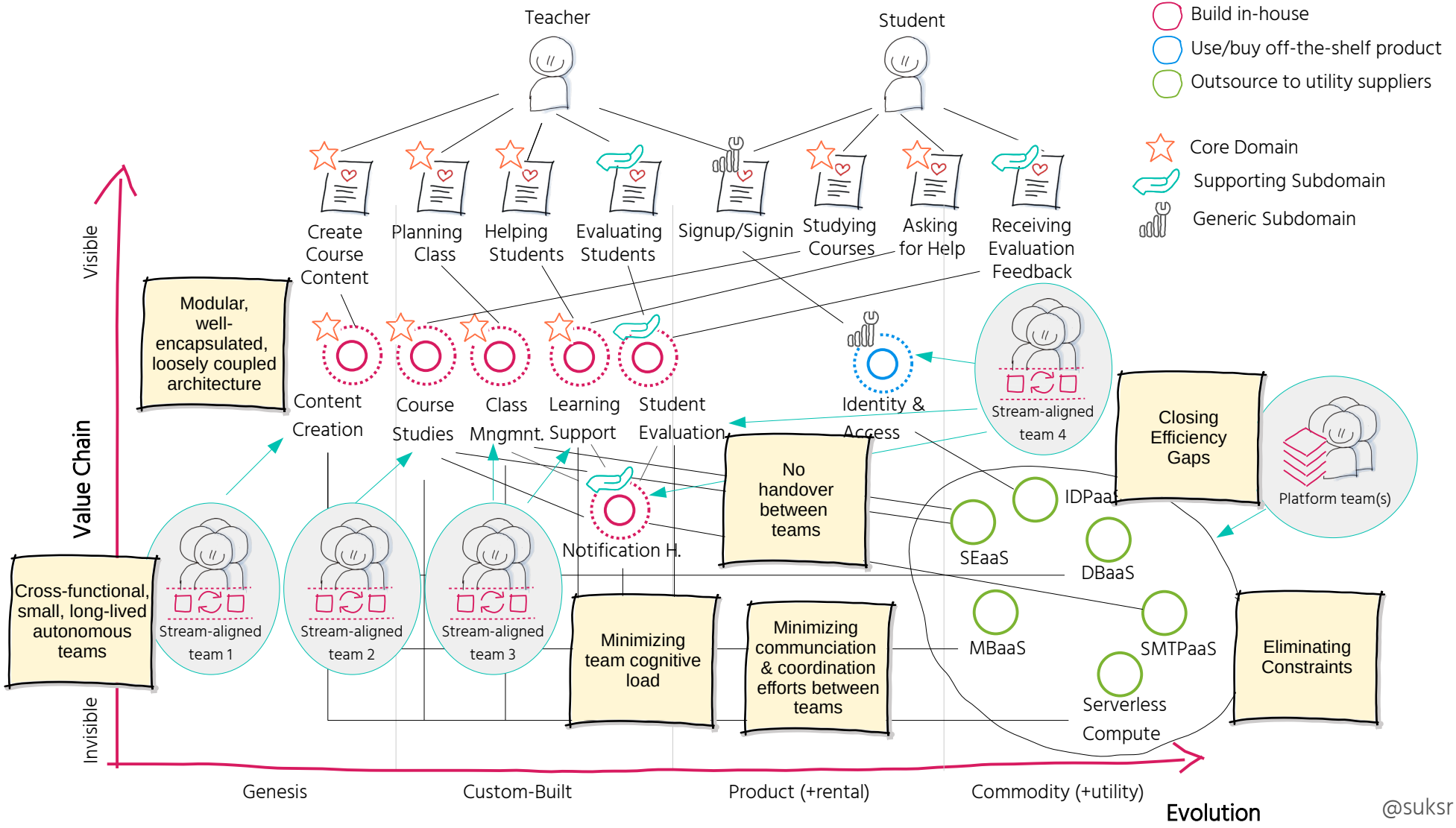
Evolution of Team Topologies



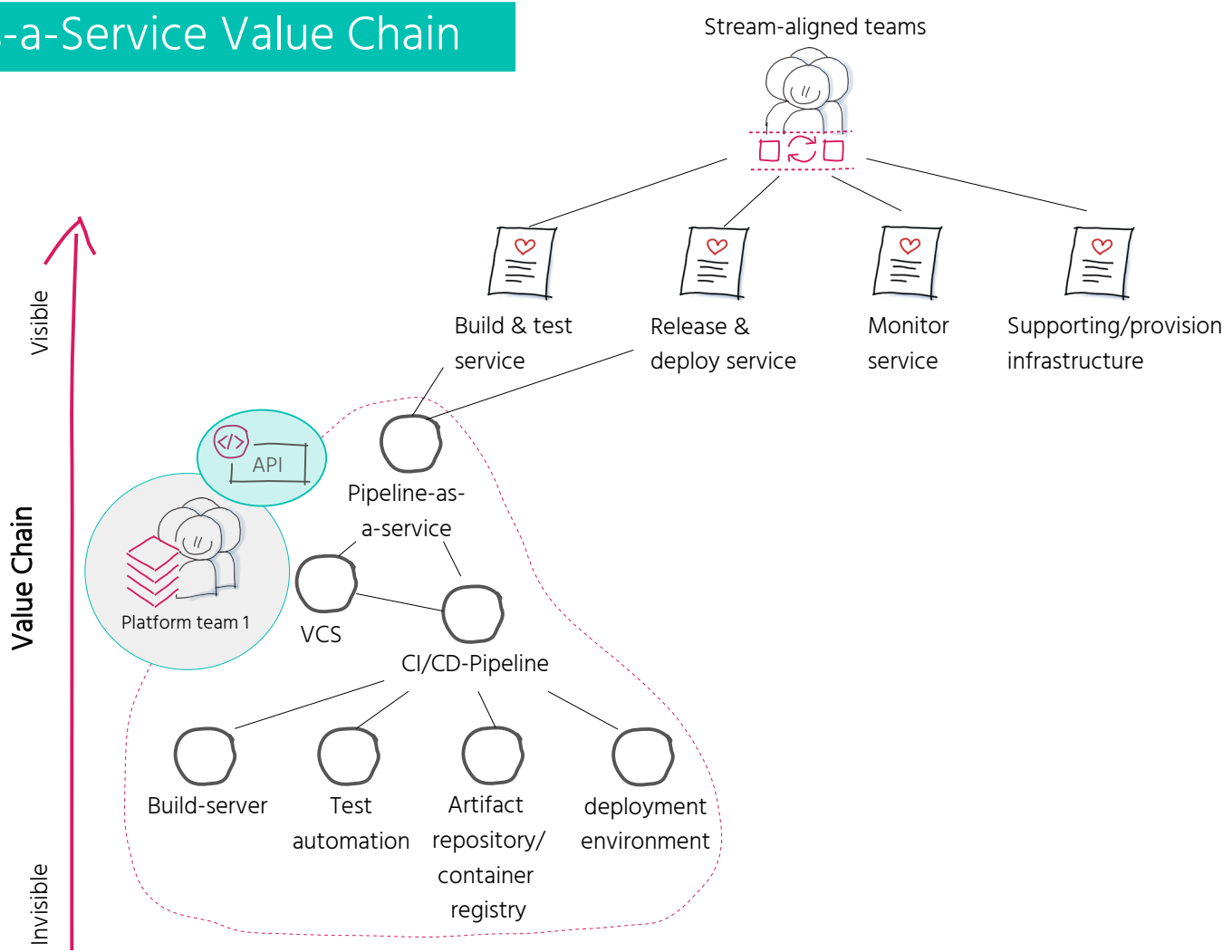




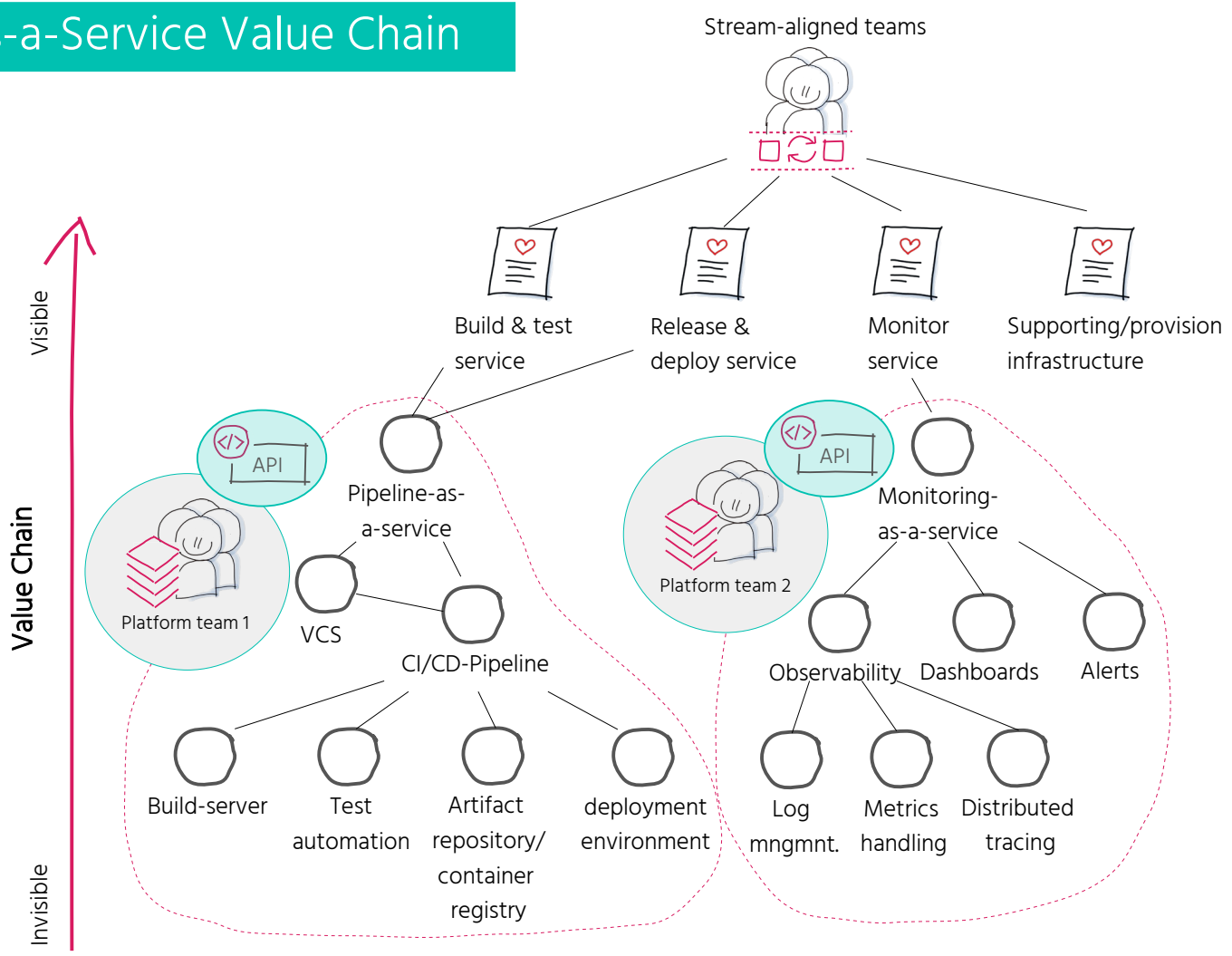
@suksr


Build in-house
Use/buy off-the-shelf product
Outsource to utility suppliers
Core Domain
Supporting Subdomain
Generic Subdomain
Teacher
Student
Create Course Content
Planning Class
Helping Students
Evaluating Students
Signup/Signin
Studying Courses
Asking for Help
Receiving Evaluation Feedback
Visible
Value Chain
Invisible
Modular, well-encapsulated, loosely coupled architecture
Content Creation
Course Studies
Class Mngmnt.
Learning Support
Student Evaluation
Notification H.
Identity & Access
Stream-aligned team 4
Closing Efficiency Gaps
Platform team(s)
No handover between teams
IDPaaS
SEaaS
DBaaS
MBaaS
SMTPaaS
Serverless Compute
Eliminating Constraints
Cross-functional, small, long-lived autonomous teams
Stream-aligned team 1
Stream-aligned team 2
Stream-aligned team 3
Minimizing team cognitive load
Minimizing communciation & coordination efforts between teams
Genesis
Custom-Built
Product (+rental)
Commodity (+utility)
Evolution
@suksr

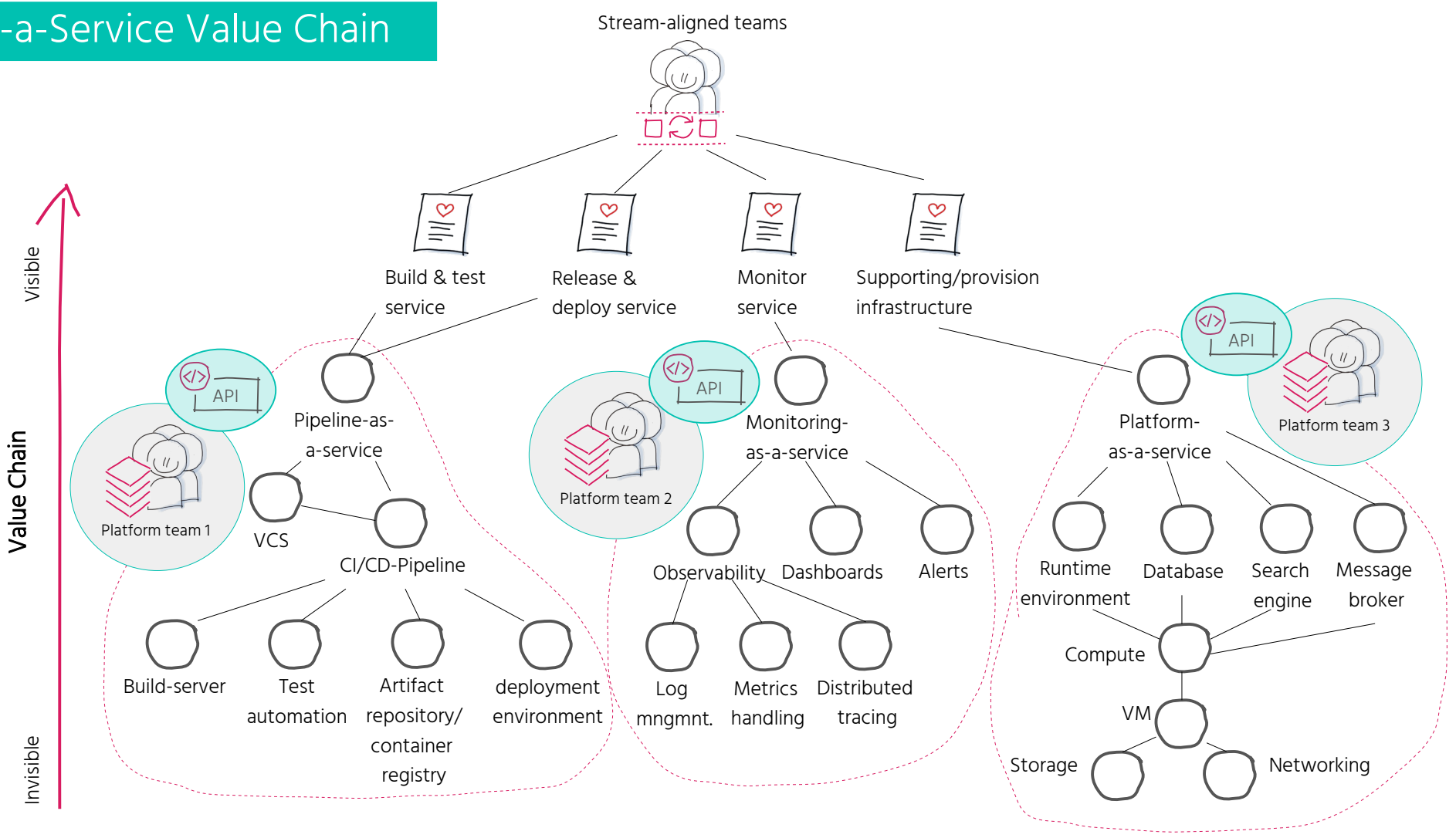
X-as-a-Service Value Chain



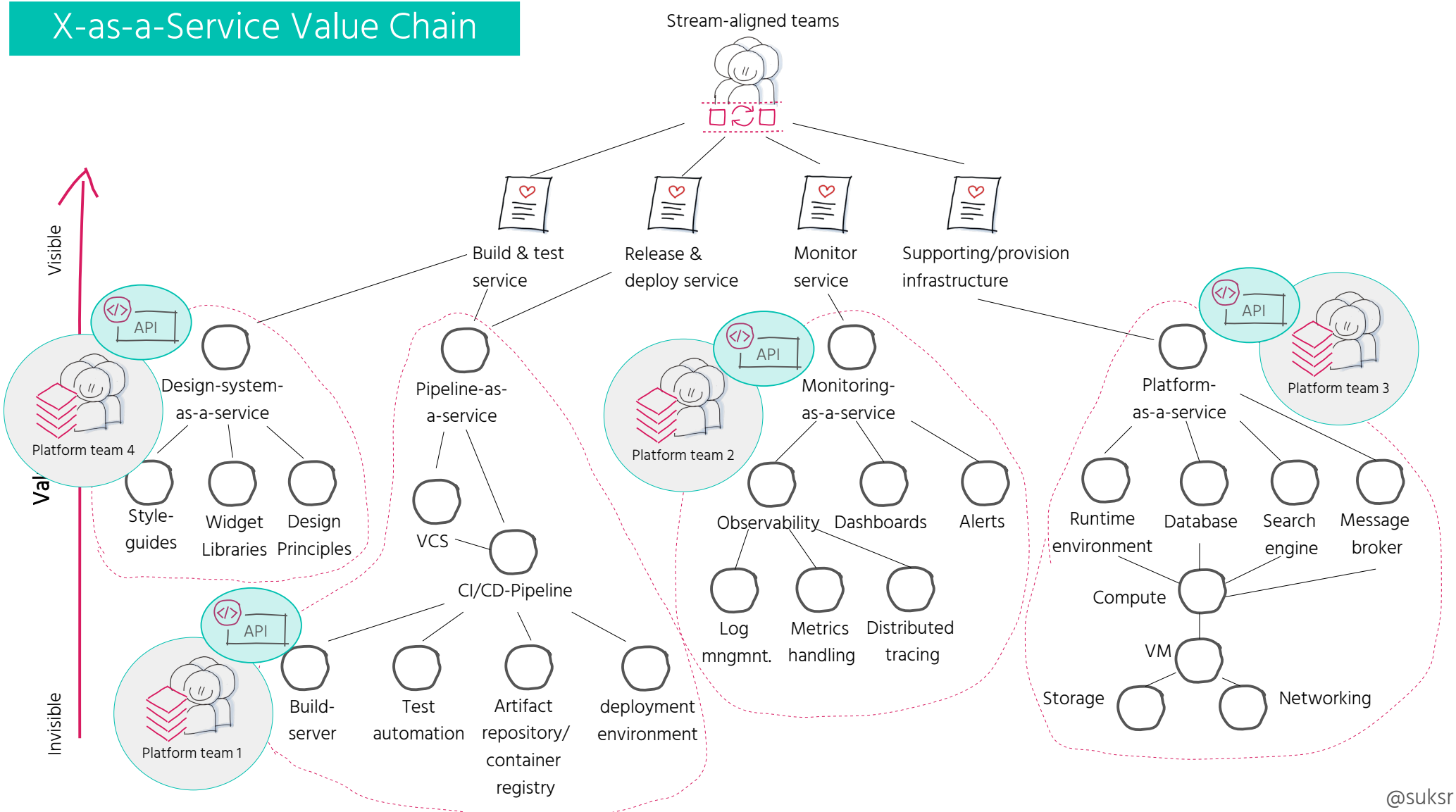
X-as-a-Service Value Chain



X-as-a-Service Value Chain

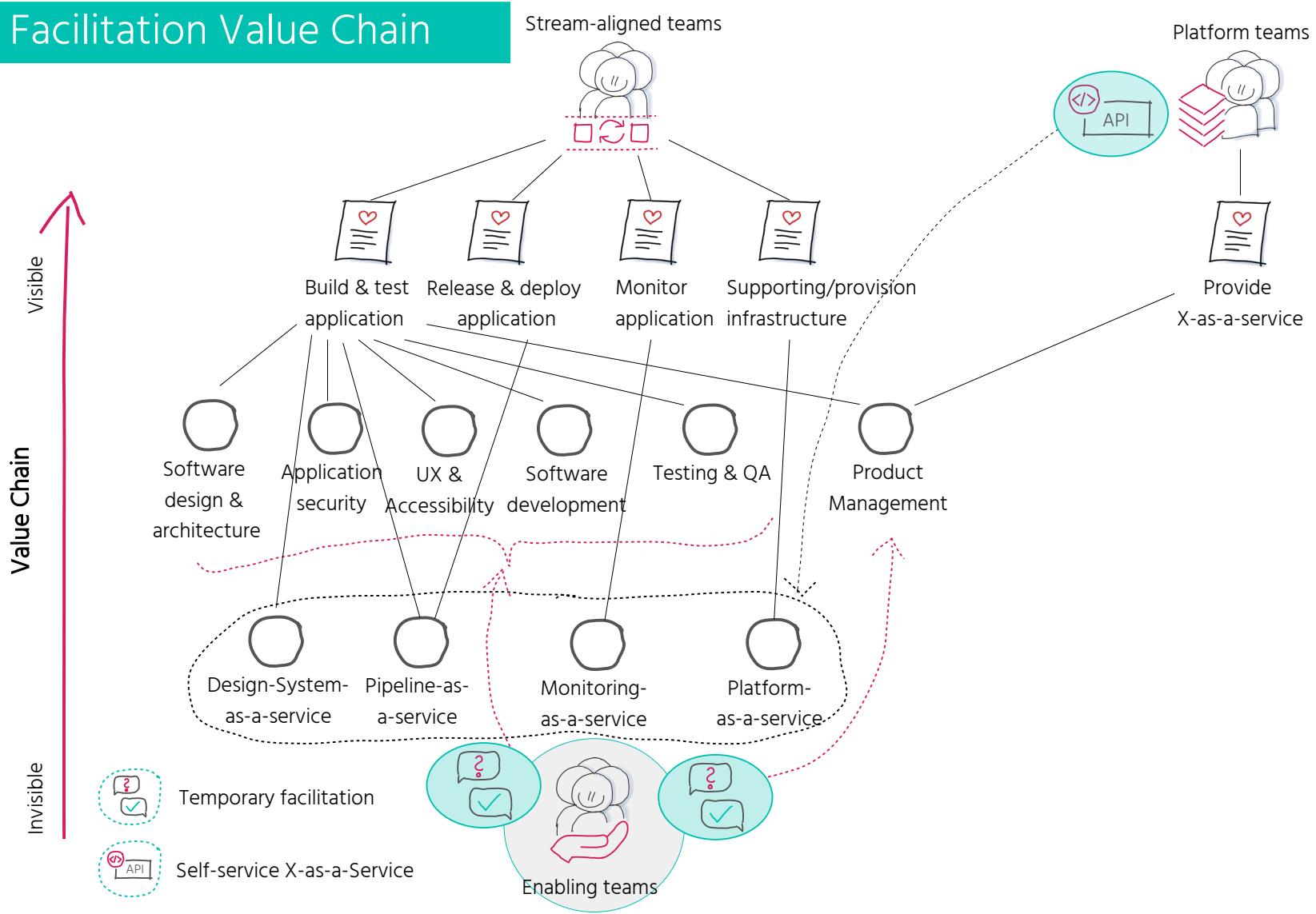


X-as-a-Service Value Chain

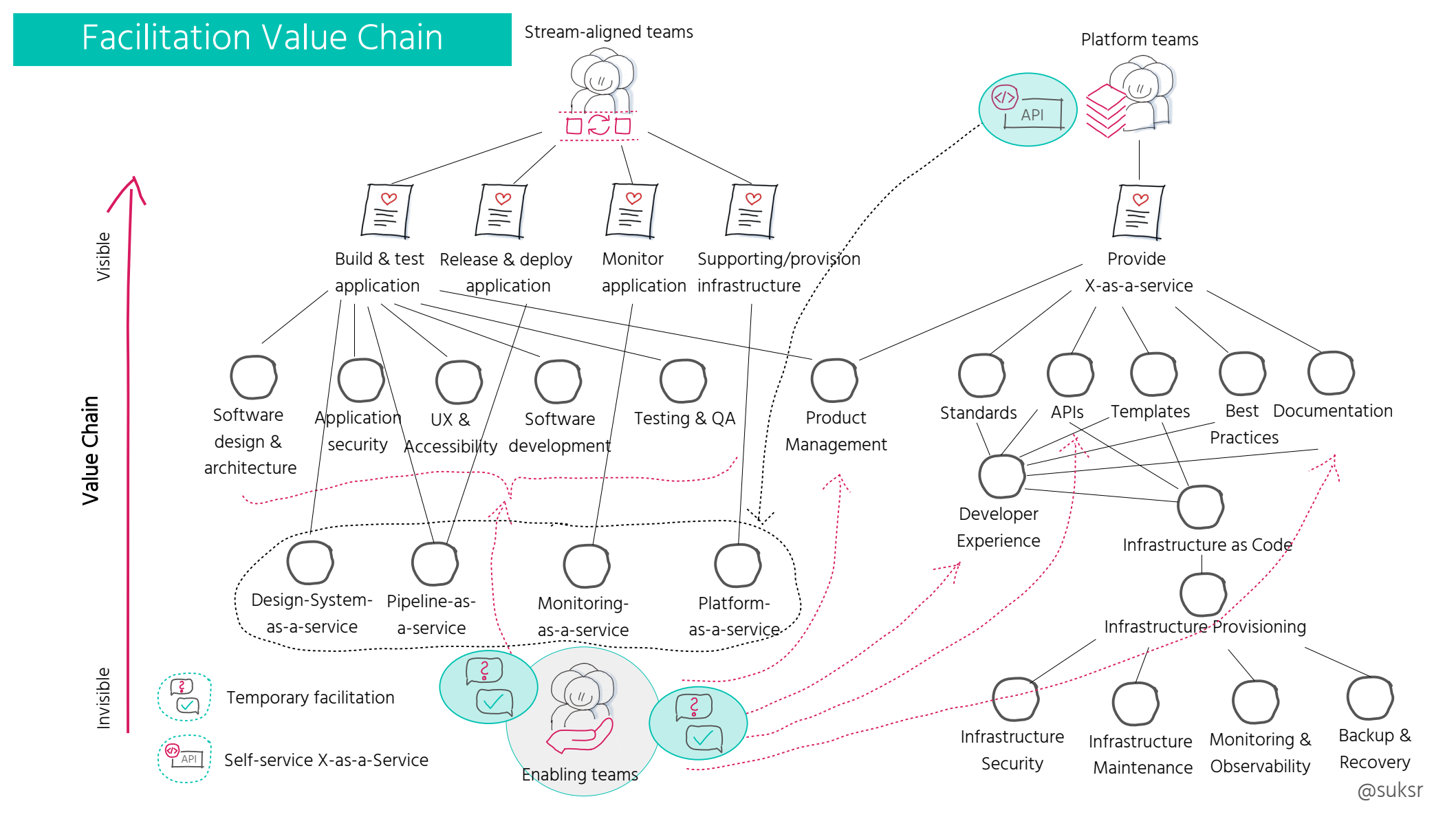


@suksr

Facilitation Value Chain

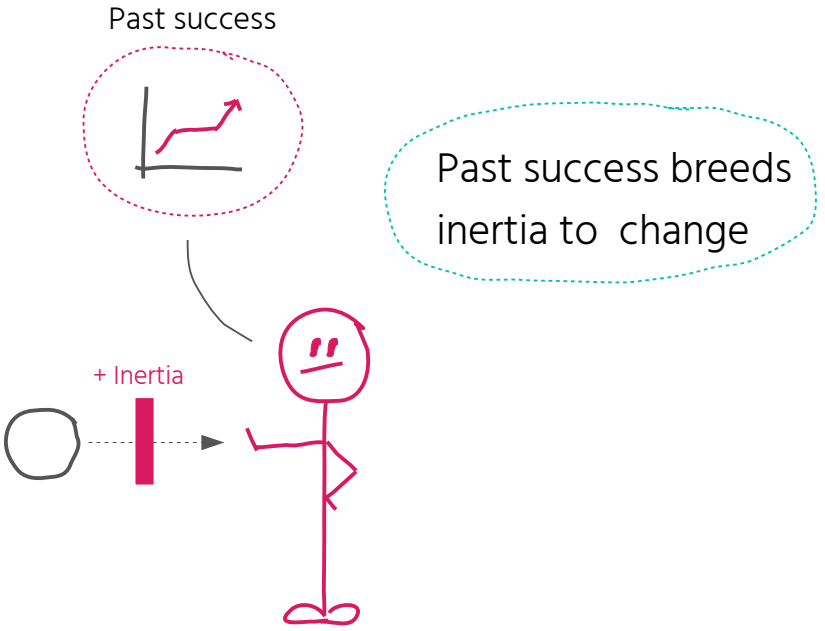


Facilitation Value Chain

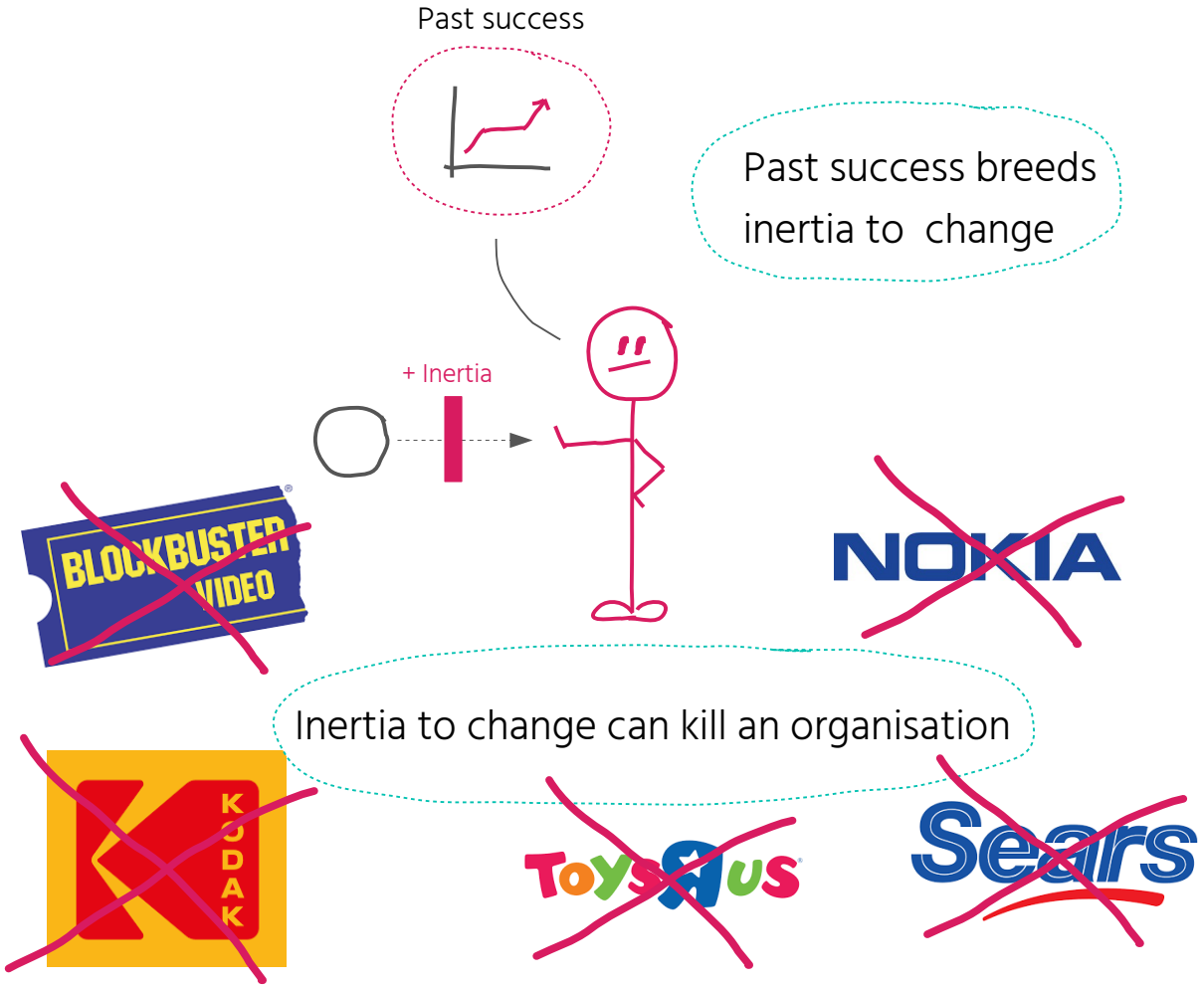


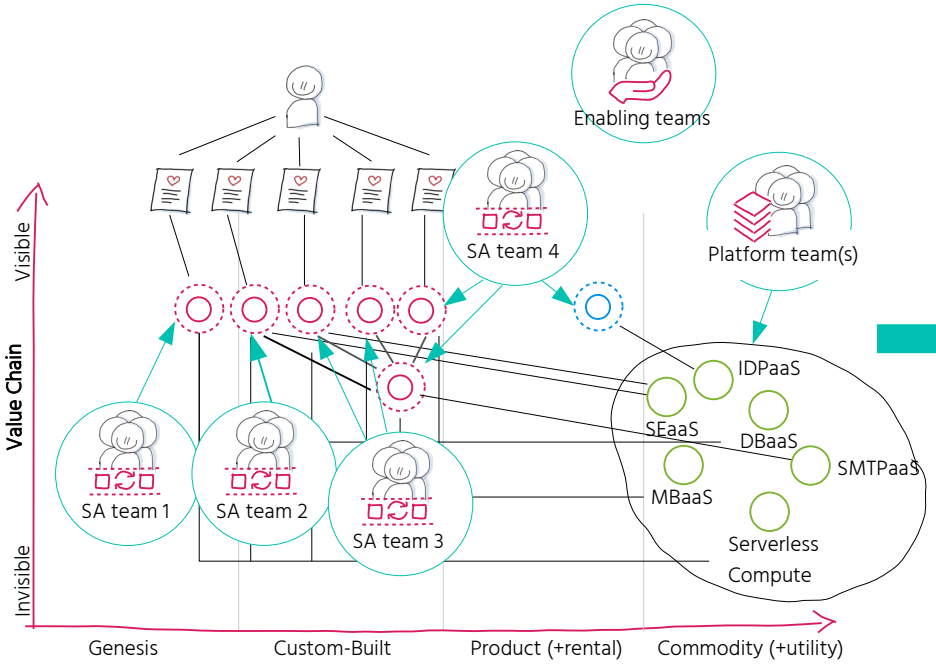
@suksr

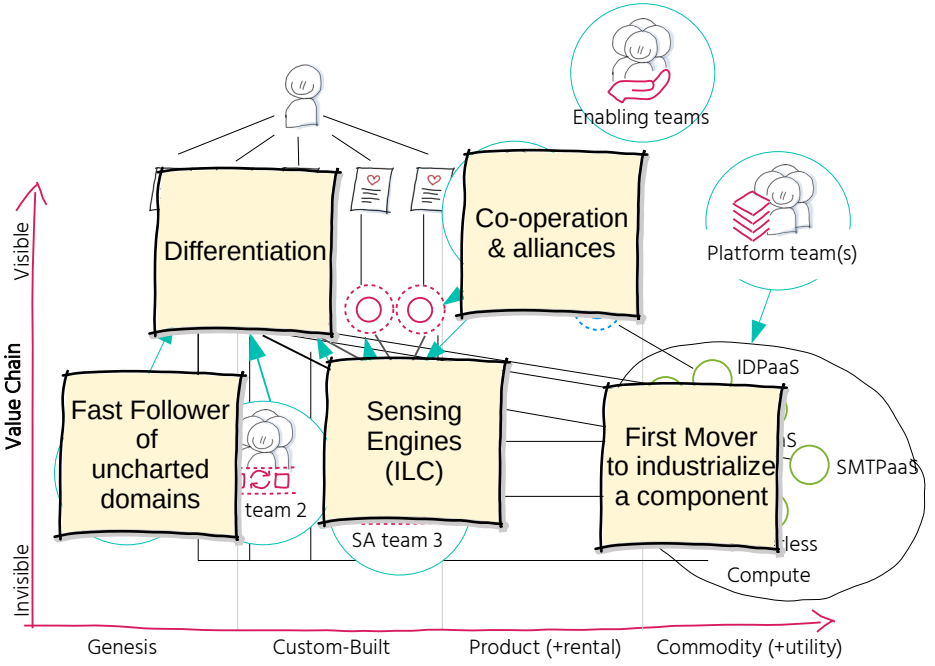
But we are successful, why should we change?



But we are successful, why should we change?





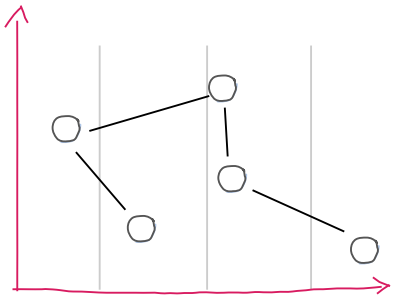


Responsive to Change

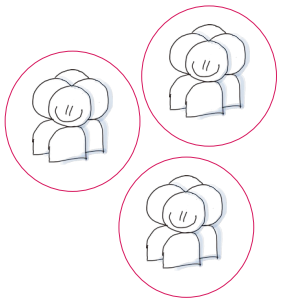
Leading future Change

Start small

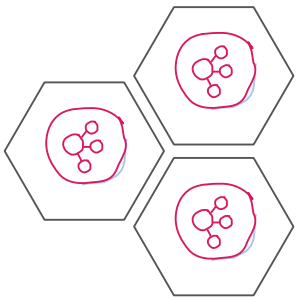
You could start with ...



... a Wardley Map
to understand your competitive landscape, and use the map as a foundation for future discussions



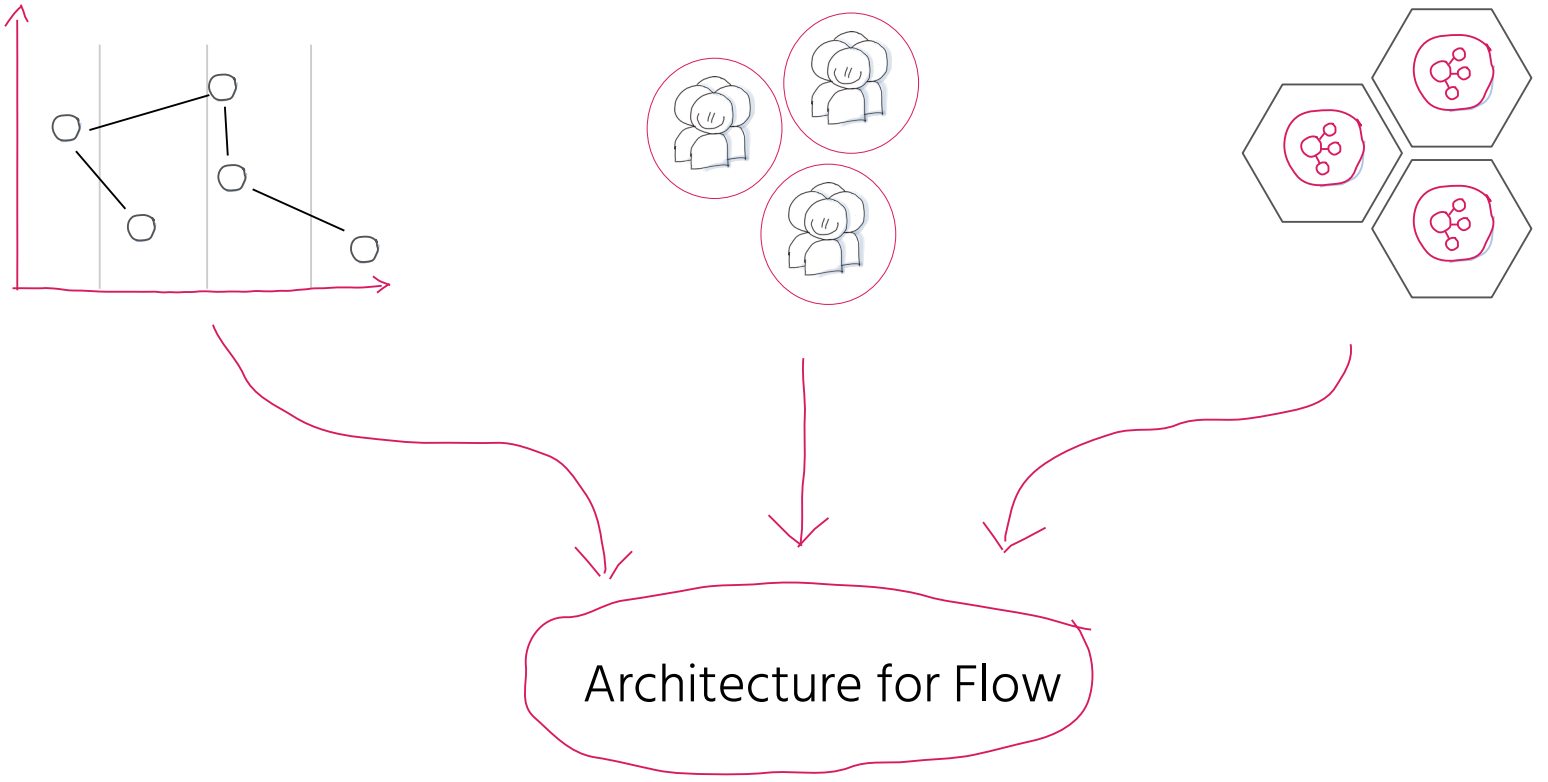
... or your current teams
and analyse their current cognitive load and bottlenecks



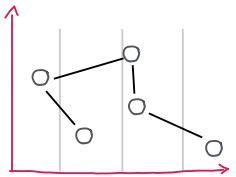
... or identifying suitables seams for modularisation
with subdomains & bounded contexts

Start small

And their paths eventually cross …

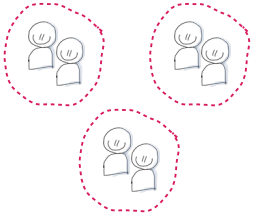


Key Takeaways



Wardley Mapping

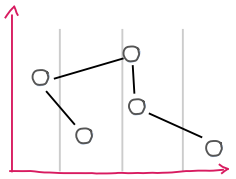
Domain-Driven Design



Team Topologies

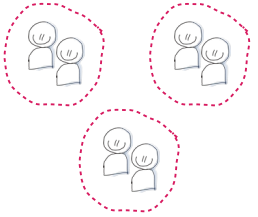
- Understanding the landscape an organization is operating & competing in including the external forces acting on the landscape
- Anticipate changes and identifying potential points to invest
- Gaining domain knowledge & discovering the core
- Knowing what components to build, buy/use, or outsource

Key Takeaways



Wardley Mapping

Domain-Driven Design

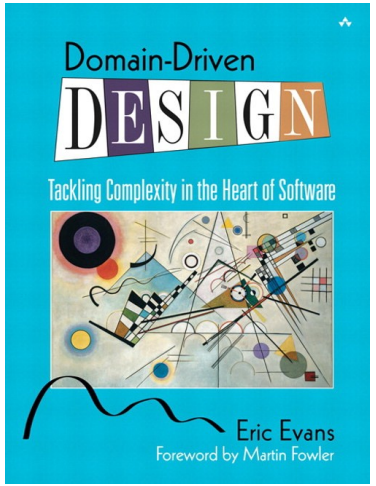


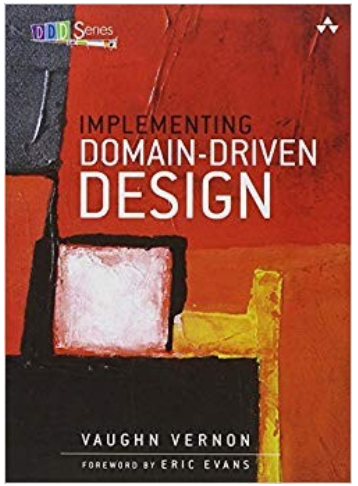
Team Topologies

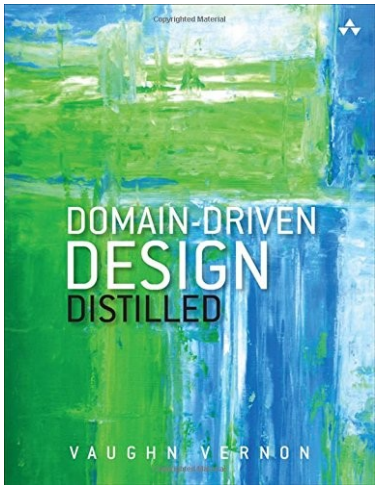
- Understanding the landscape an organization is operating & competing in including the external forces acting on the landscape
- Anticipate changes and identifying potential points to invest
- Gaining domain knowledge & discovering the core
- Knowing what components to build, buy/use, or outsource

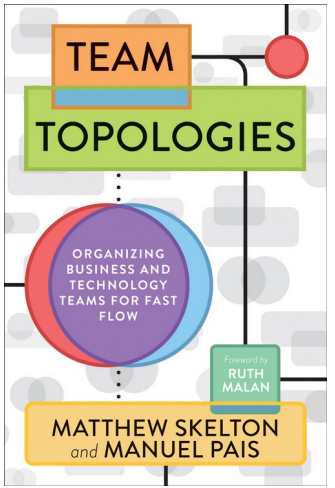
- Modular, well-encapsulated architecture & well-defined ownership boundaries w/ bounded contexts
- Aligning teams and evolving their interactions to the system we build & the strategy we plan
- Eliminating delivery bottlenecks & increasing software delivery performance
- Optimizing for a a fast flow of change to be able to adapt, evolve and thrive in the face of constant change

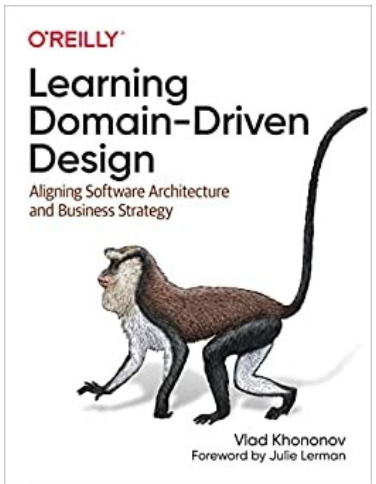
Some References

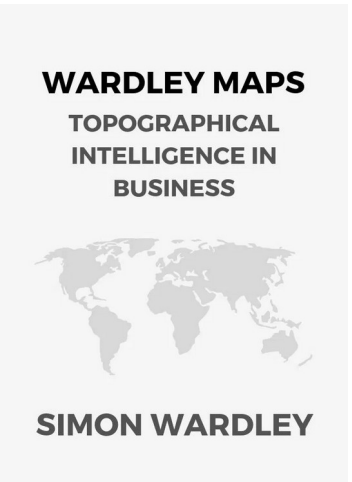


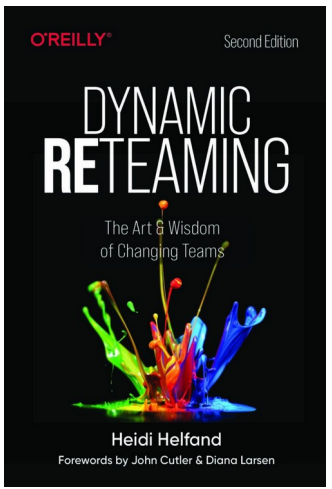






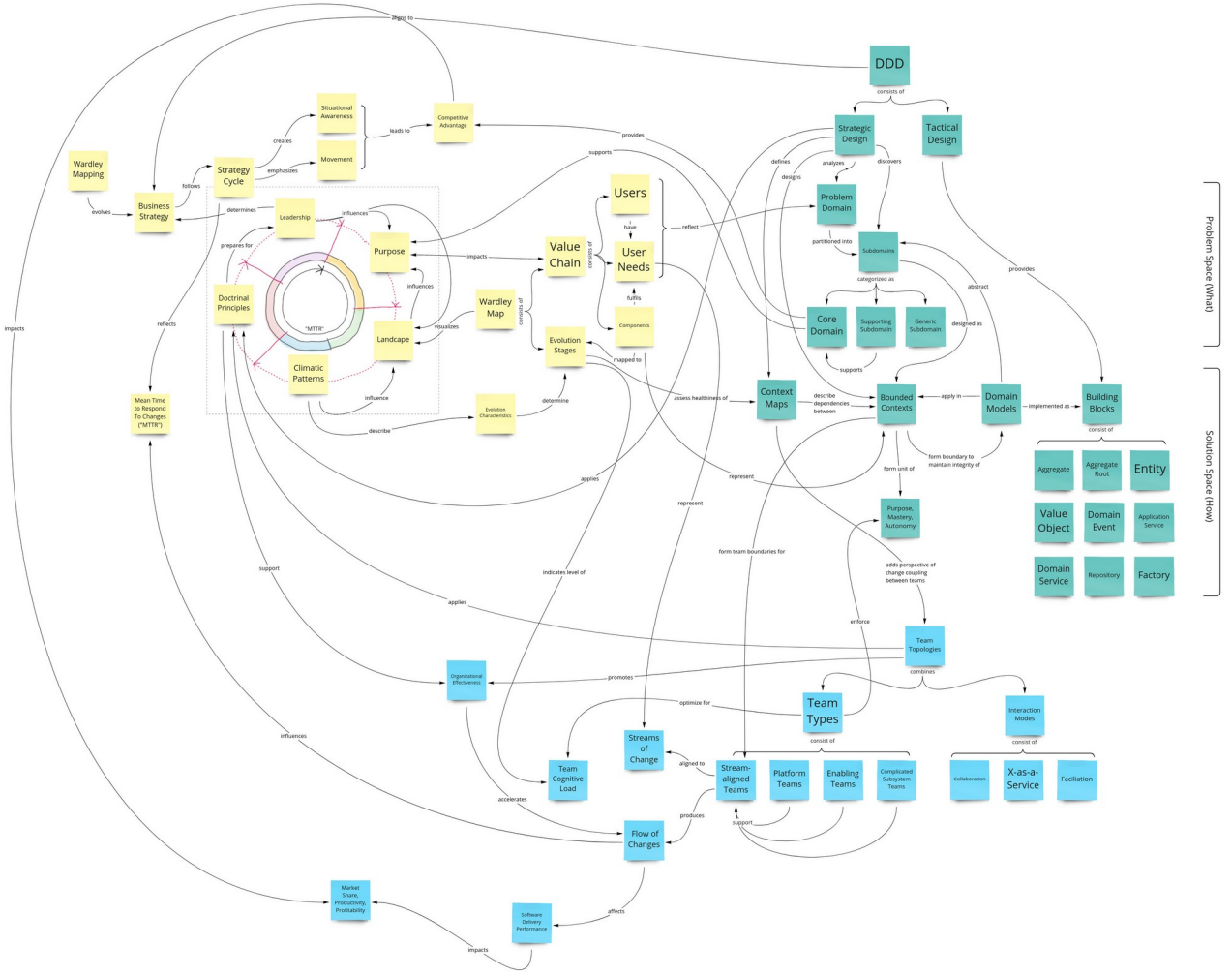


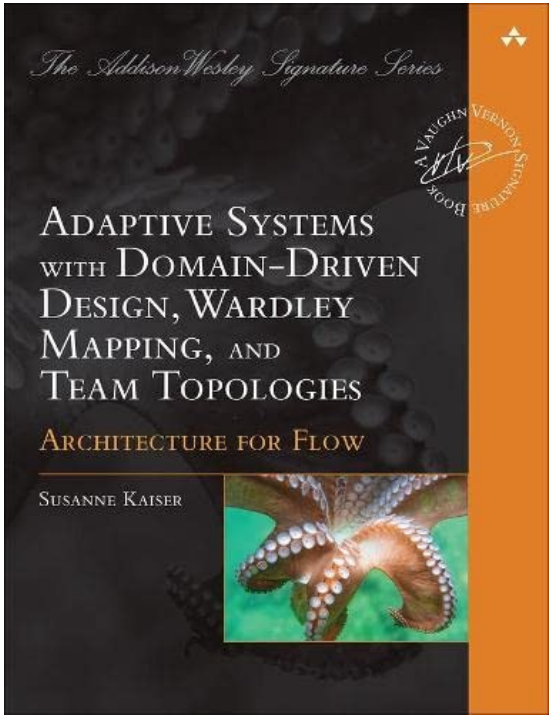




https://medium.com/wardleymaps
https://learnwardleymapping.com/
https://github.com/wardley-maps-community/awesome-wardley-maps
https://github.com/ddd-crew
https://www.dddheuristics.com

If you are interested in more details ...





THANK YOU

Susanne Kaiser
Independent Tech Consultant
@suksr