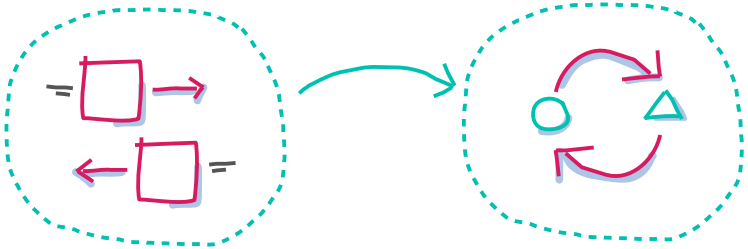


Building Adaptive Systems for a Fast Flow of Change

Susanne Kaiser
Independent Tech Consultant
@suksr

Example: Nokia

Nokia N-Series

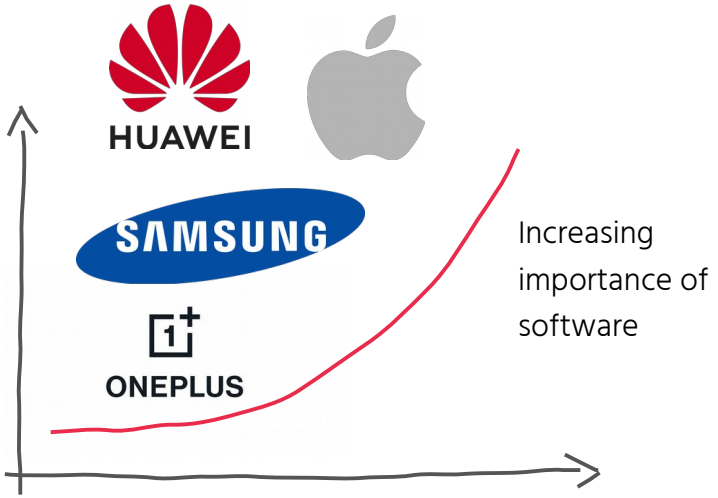
Apple iPhone

2005

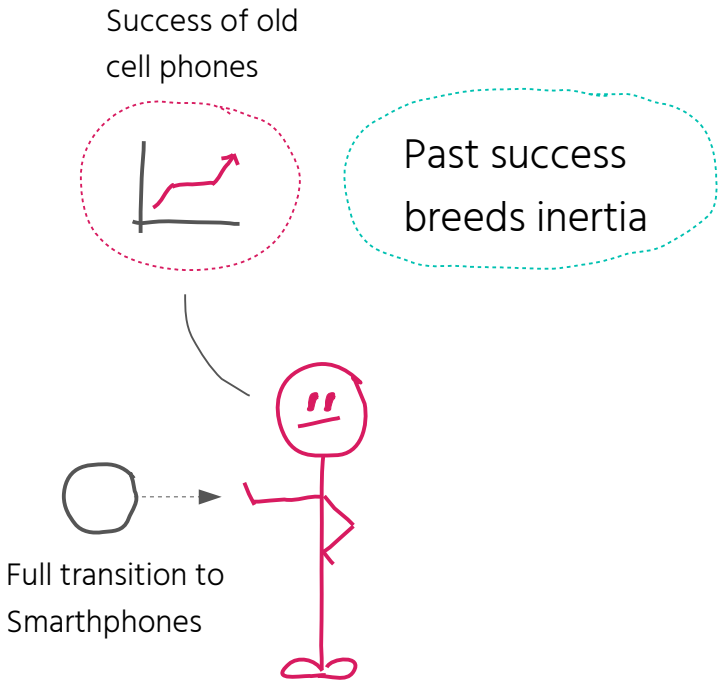
2007

@suksr

Competitors Actions And Inertia to Change



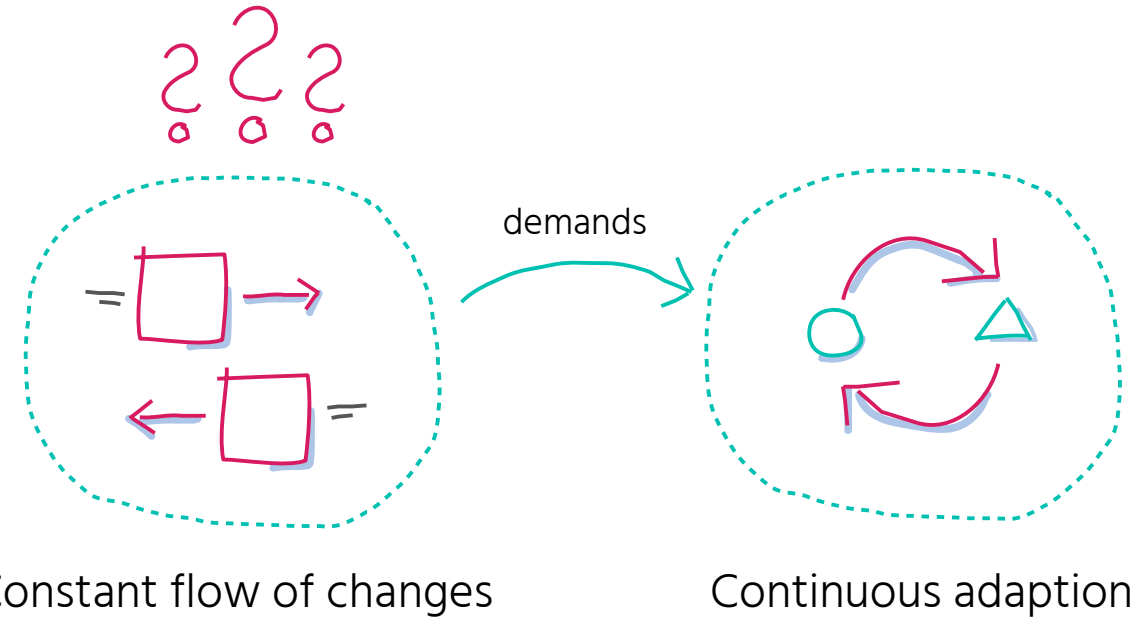
Competitors actions will change the game



Inertia can kill an organisation

@suksr

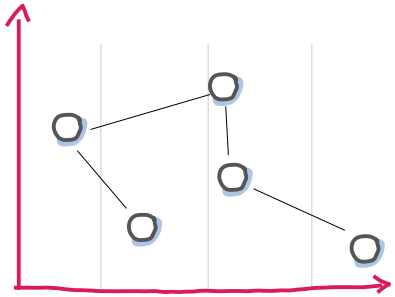
Constant Flow of Changes → Continuous Adaption



3 Perspectives to Build Adaptive Systems

1.

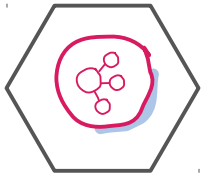
Business-Strategy



w/ Wardley Maps

2.

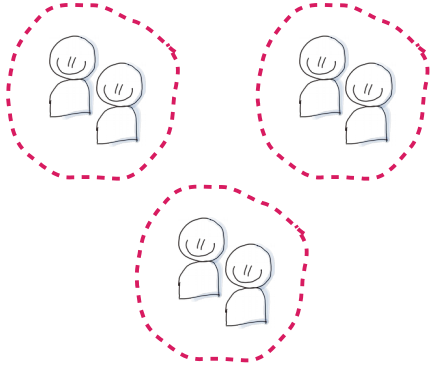
Software-Design/
-Architecture



w/ Domain-Driven Design

3.

Team-Organization



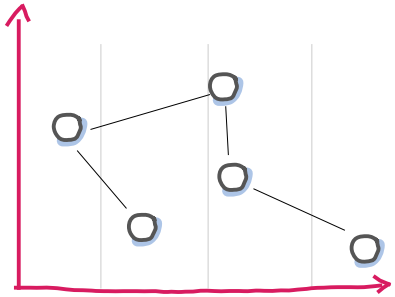
w/ Team Topologies

@suksr

3 Perspectives to Build Adaptive Systems

1.

Business-Strategy



w/ Wardley Maps

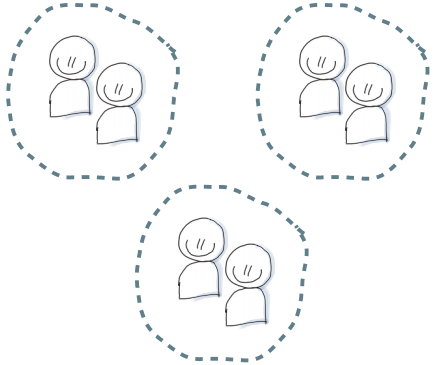
2.

Software-Design/
-Architecture

w/ Domain-Driven Design

3.

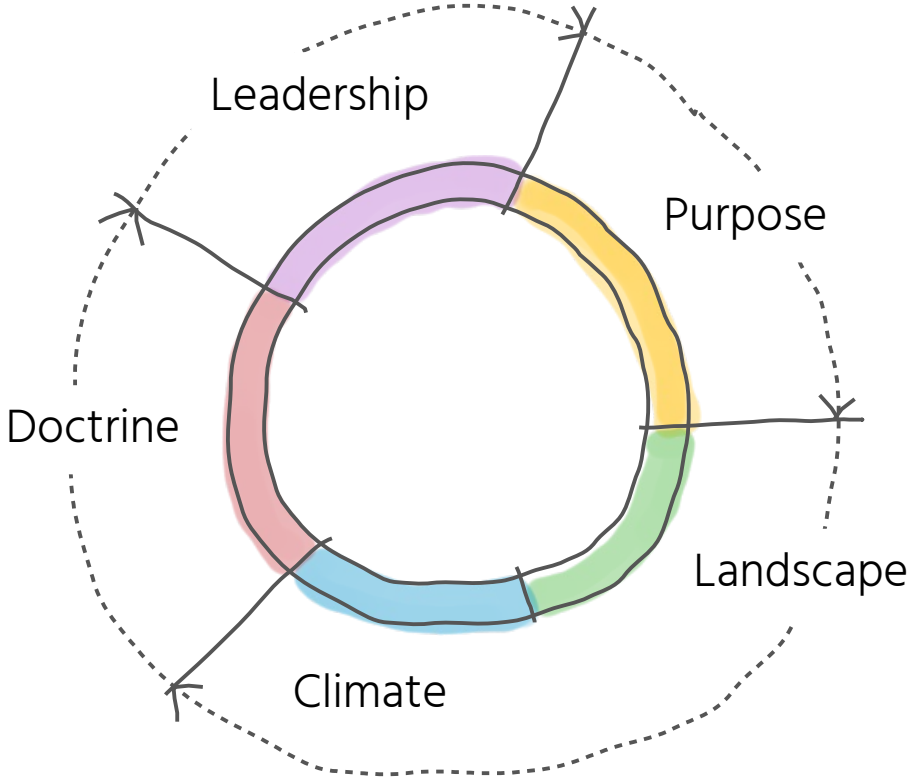
Team-Organization



w/ Team Topologies

@suksr

Business Strategy w/ Wardley Maps



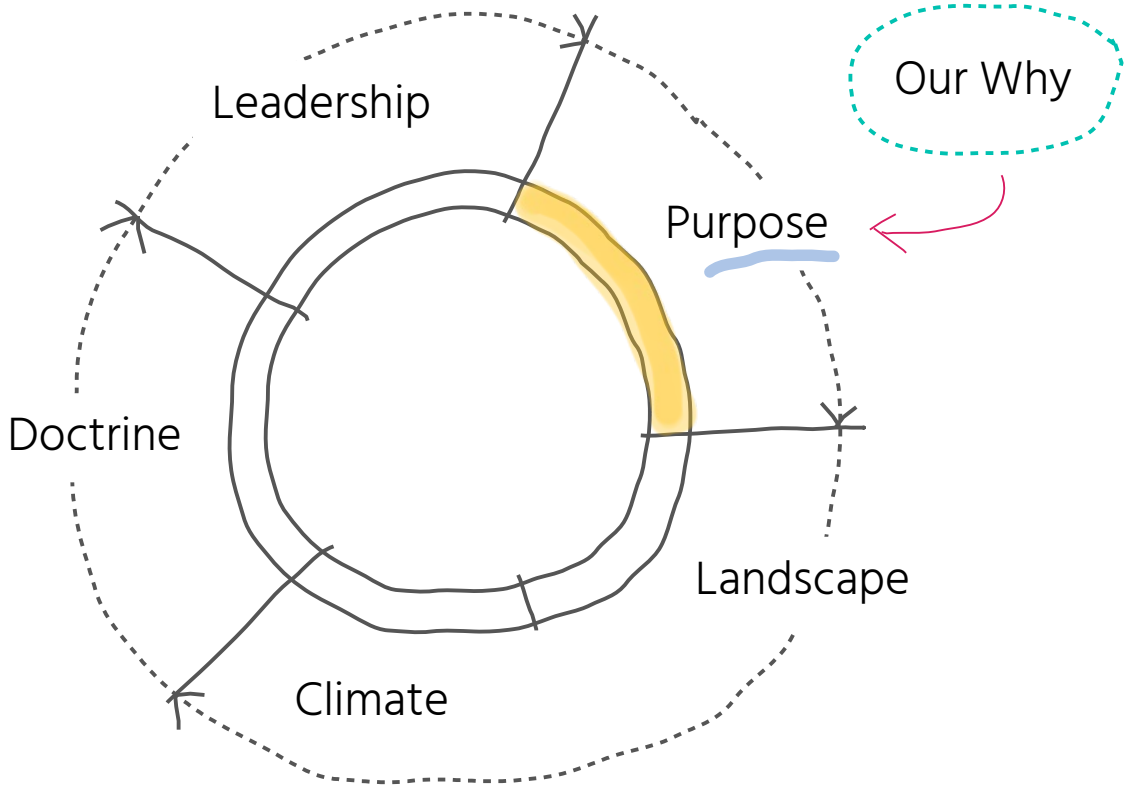
"The strategy cycle is a representation of change and how we need to react to it."

Simon Wardley

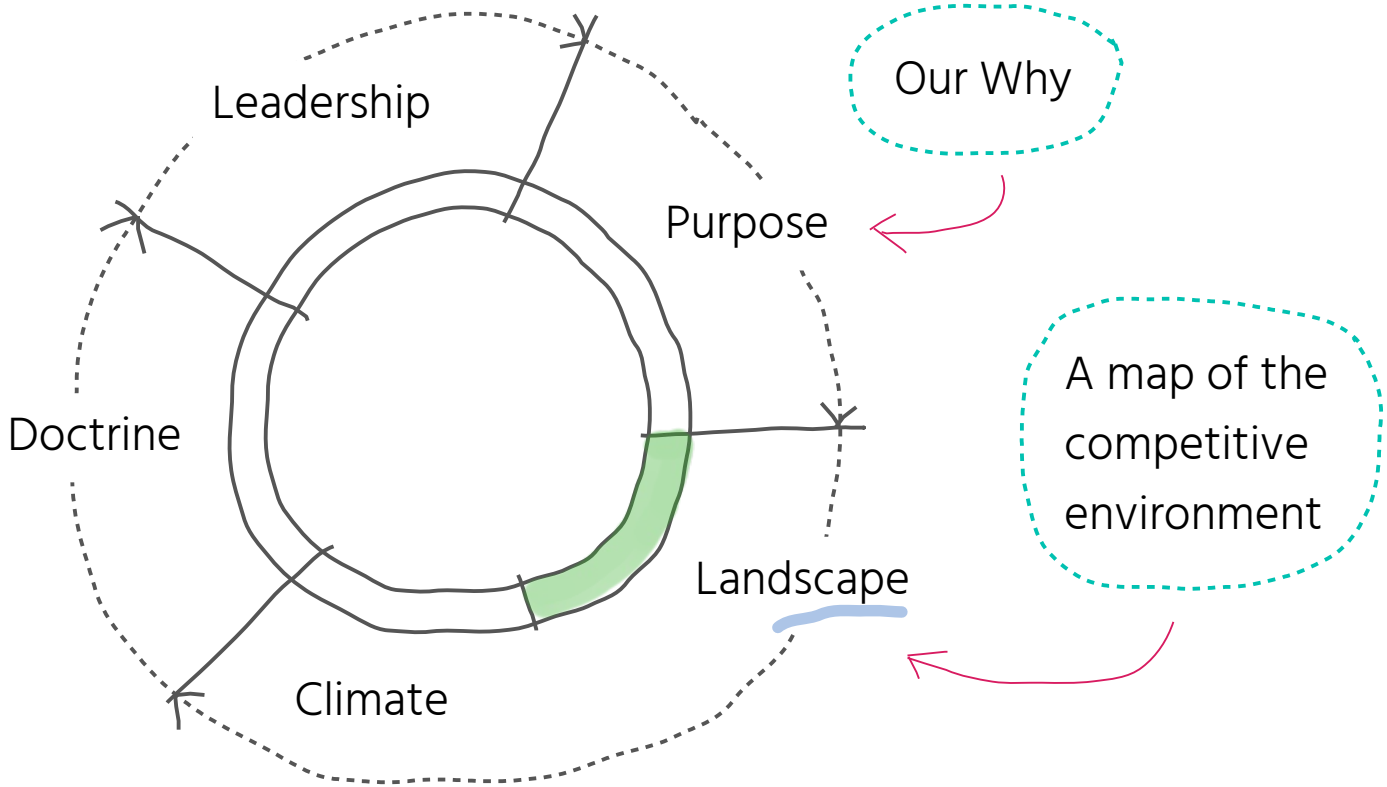
The Strategy Cycle

@suksr

Strategy Cycle

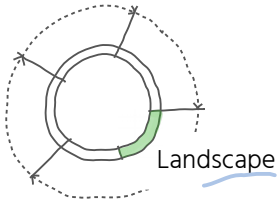


Strategy Cycle



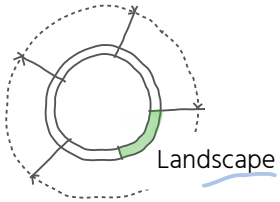
@suksr

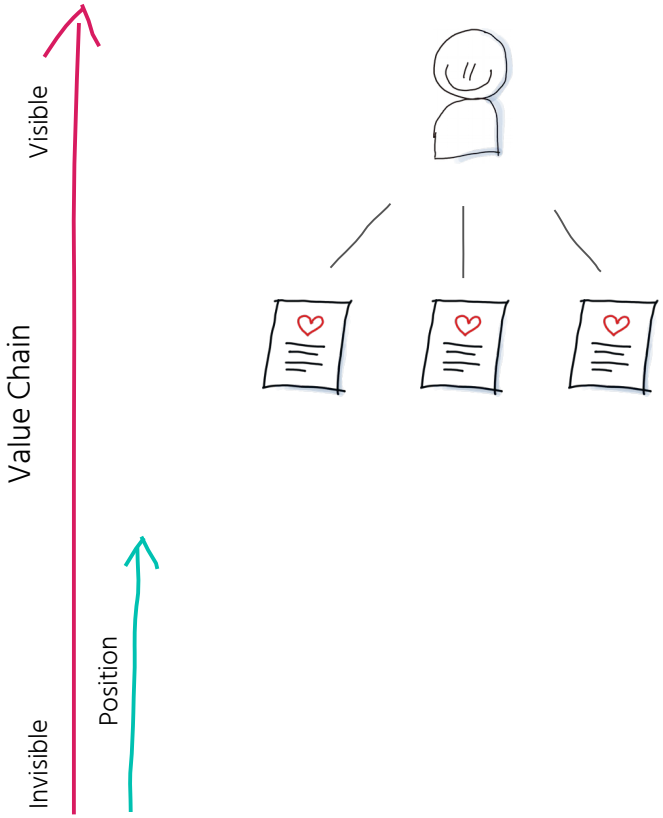
Wardley Maps – VALUE CHAIN



Who are your users?

Wardley Maps – VALUE CHAIN

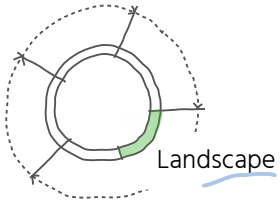


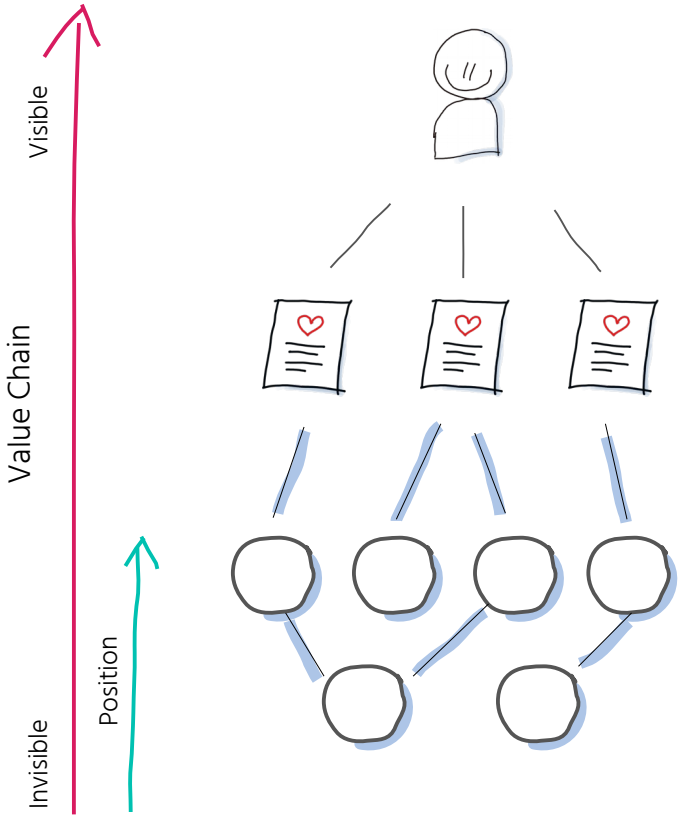


Who are your users?

What are your users' needs?

Wardley Maps – VALUE CHAIN



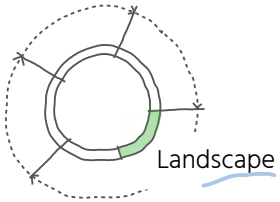


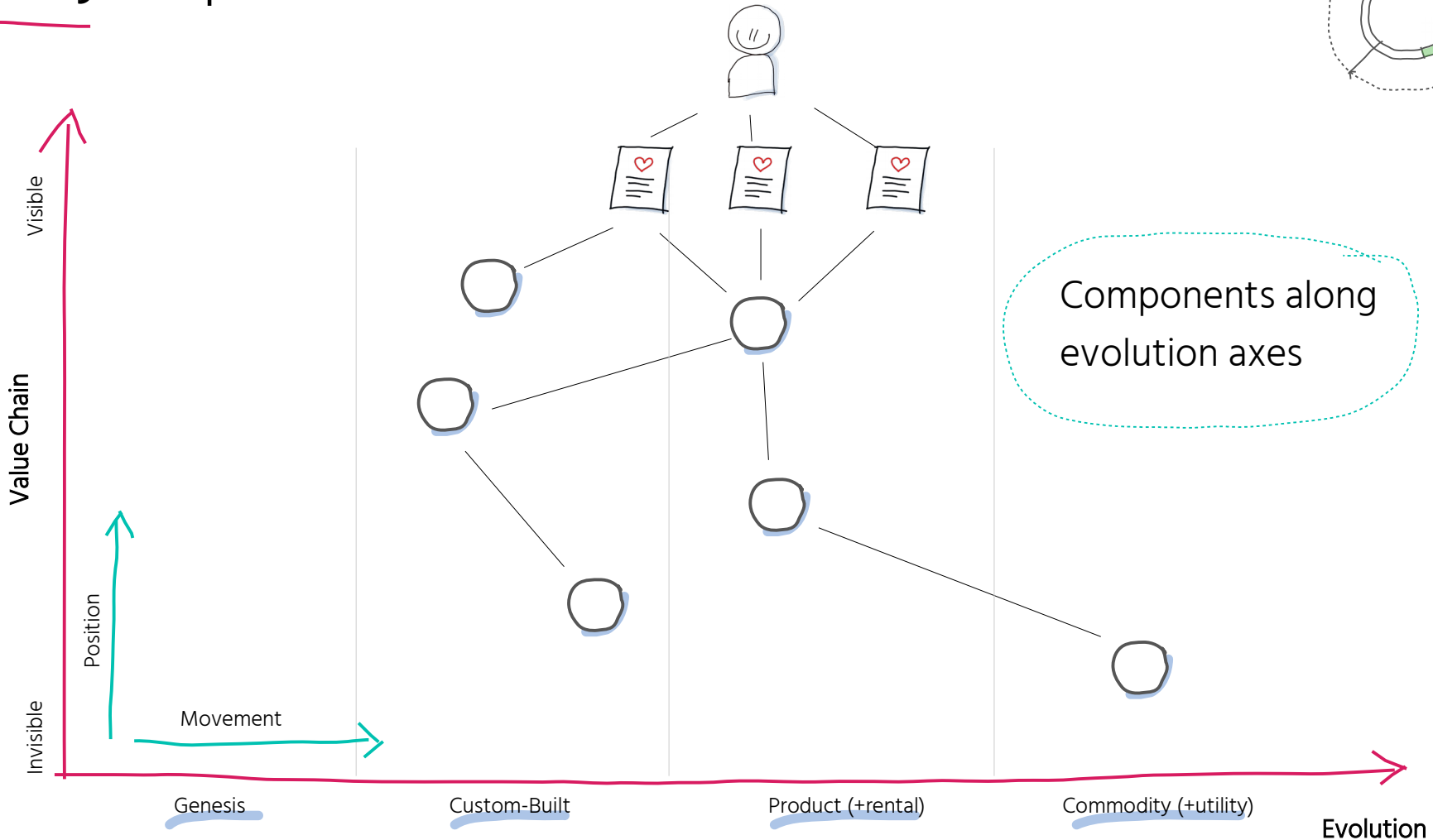
Who are your users?

What are your users' needs?

What are the components/activities to fulfill your users' needs incl. dependencies?

Wardley Maps – EVOLUTION STAGES

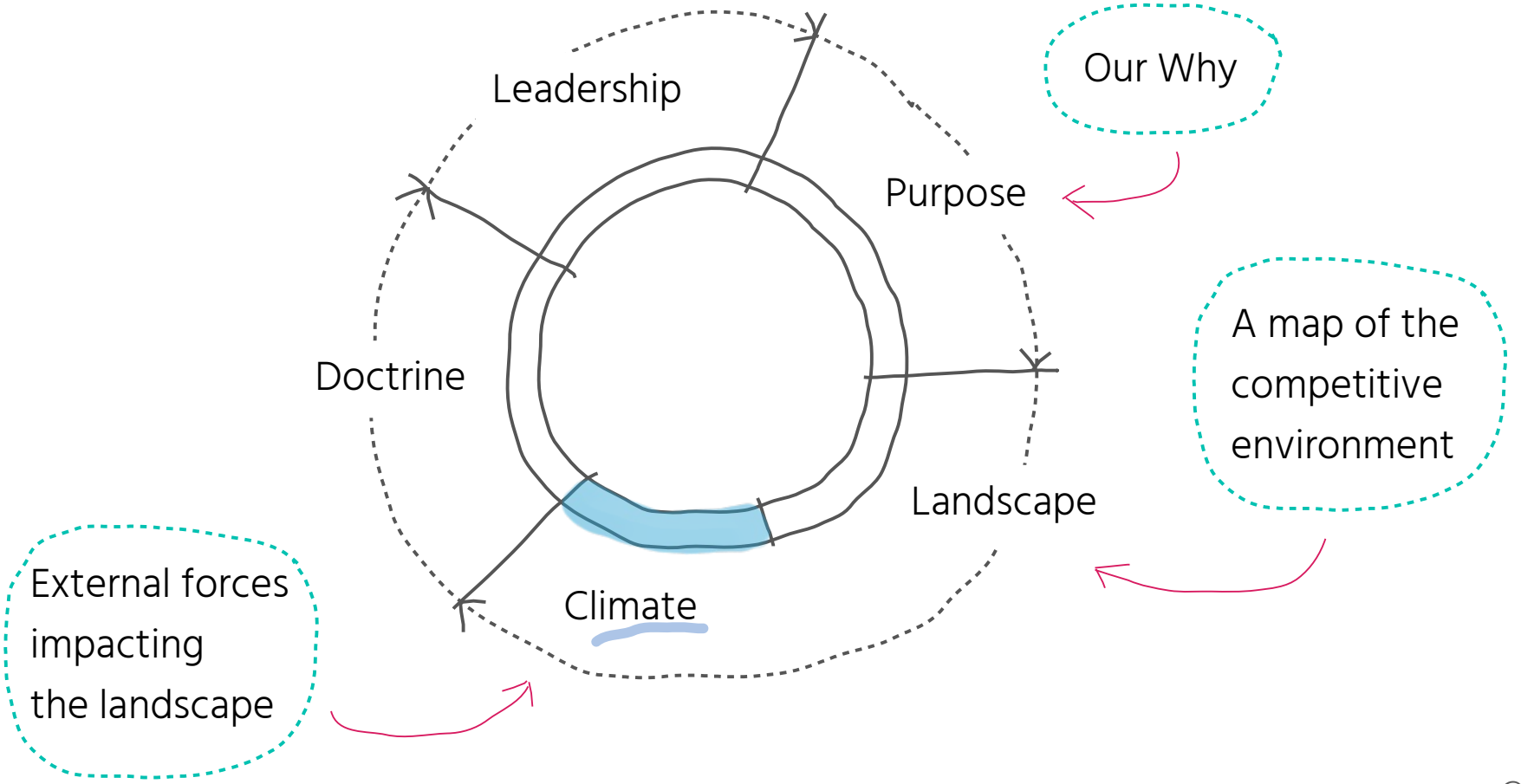




Components along evolution axes

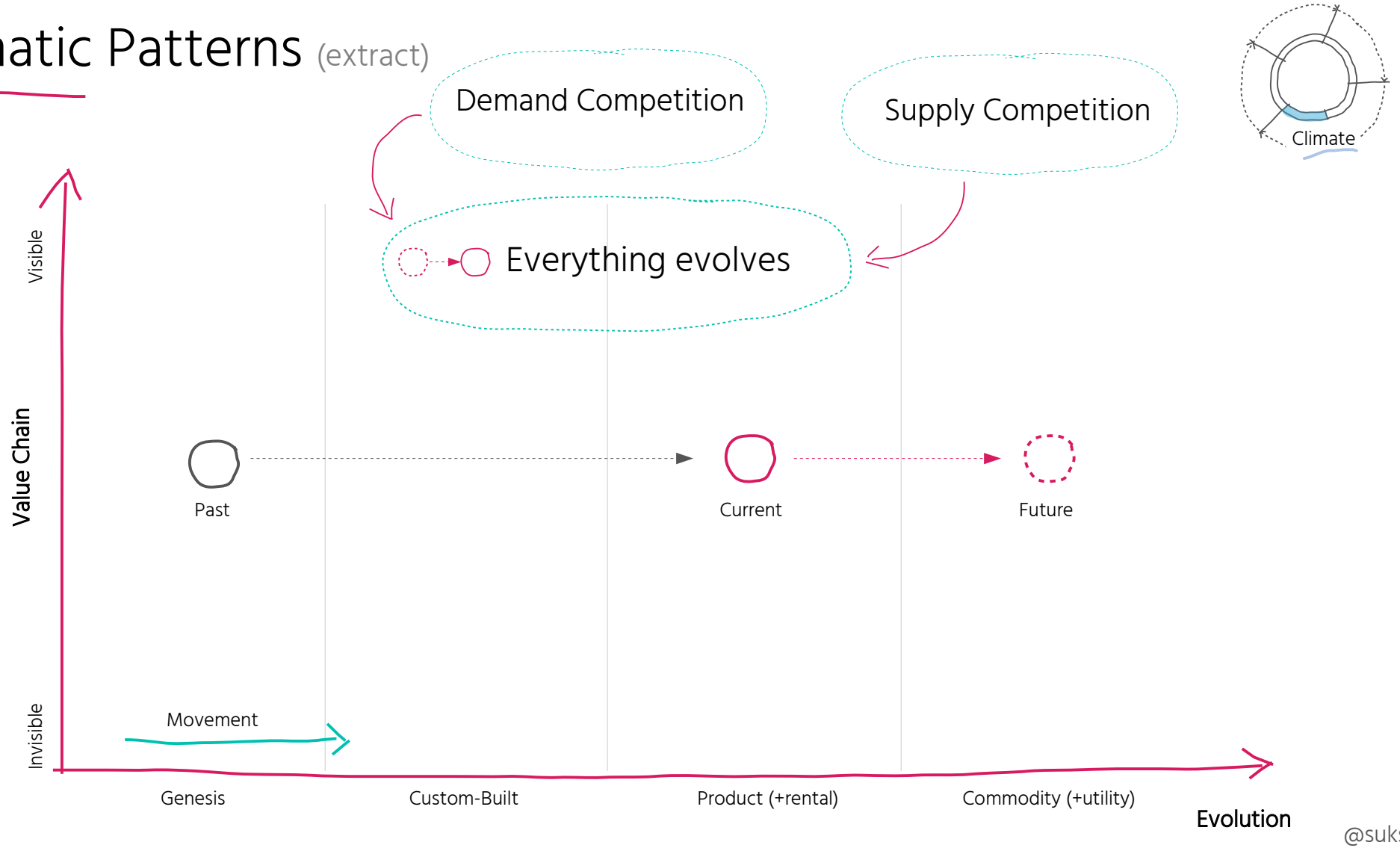
@suksr

Strategy Cycle



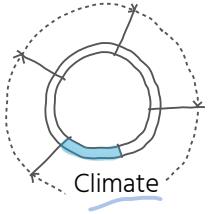
@suksr

Climatic Patterns (extract)



@suksr

Climatic Patterns (extract)



Characteristics change

Value Chain

Visible

Invisible

Uncharted		Industrialised
Undefined market Uncertain Unpredictable Rare	Forming market Learning on use Increasing understanding Slowly increasing consumption	Growing market Learning on operation Increasing education Rapidly increasing consumption	Mature market Known / accepted Stable Widespread and stabilising
Poorly understood Chaotic	Rapid increases in learning Domain of experts	Rapid increase in use Increasing expectation of use	Commonly understood Ordered

Past → Current → Future

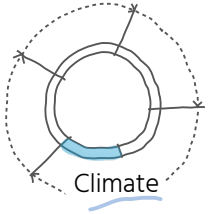
Movement

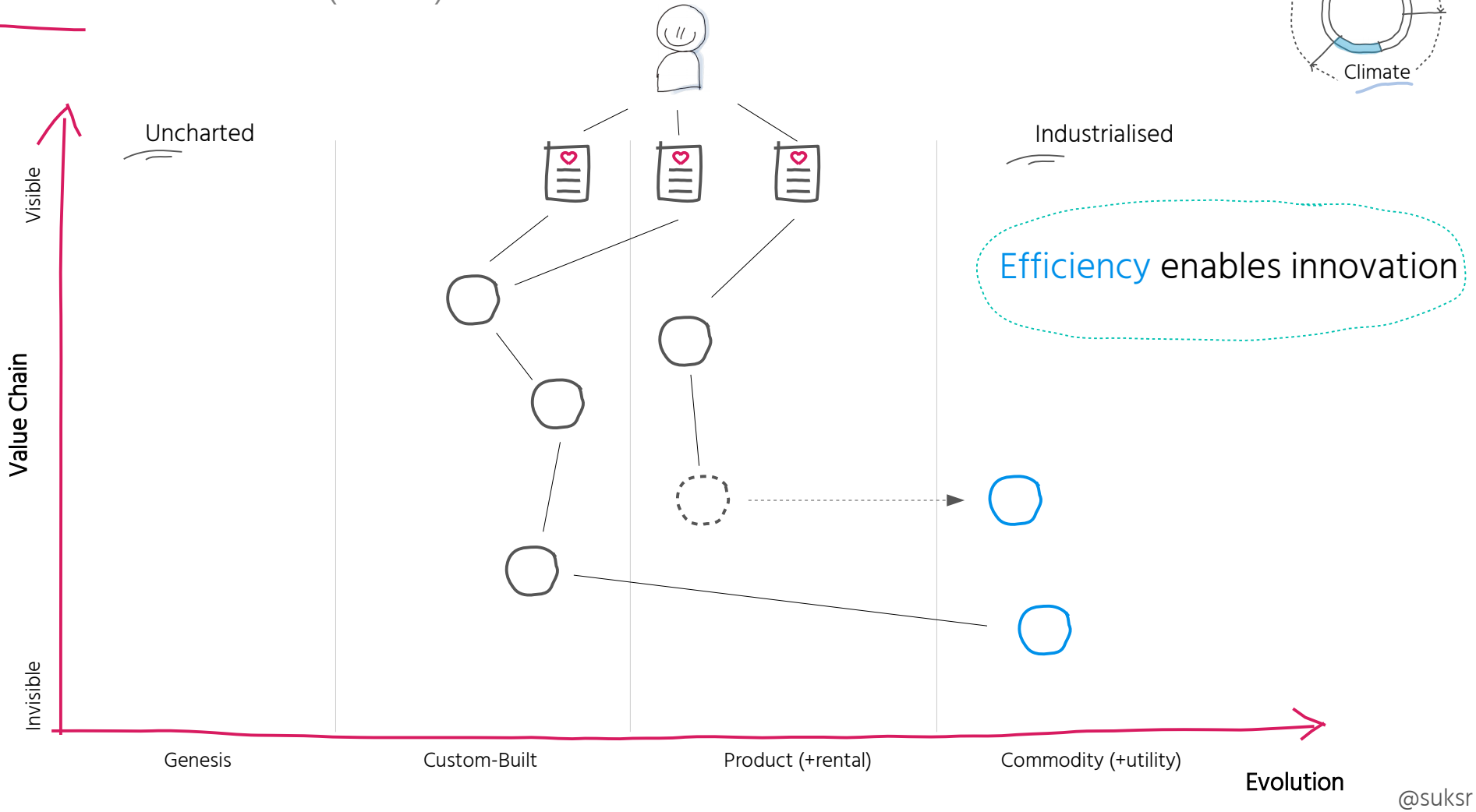
Genesis | Custom-Built | Product (+rental) | Commodity (+utility)

Evolution

@suksr

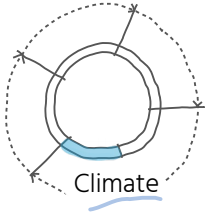
Climatic Patterns (extract)

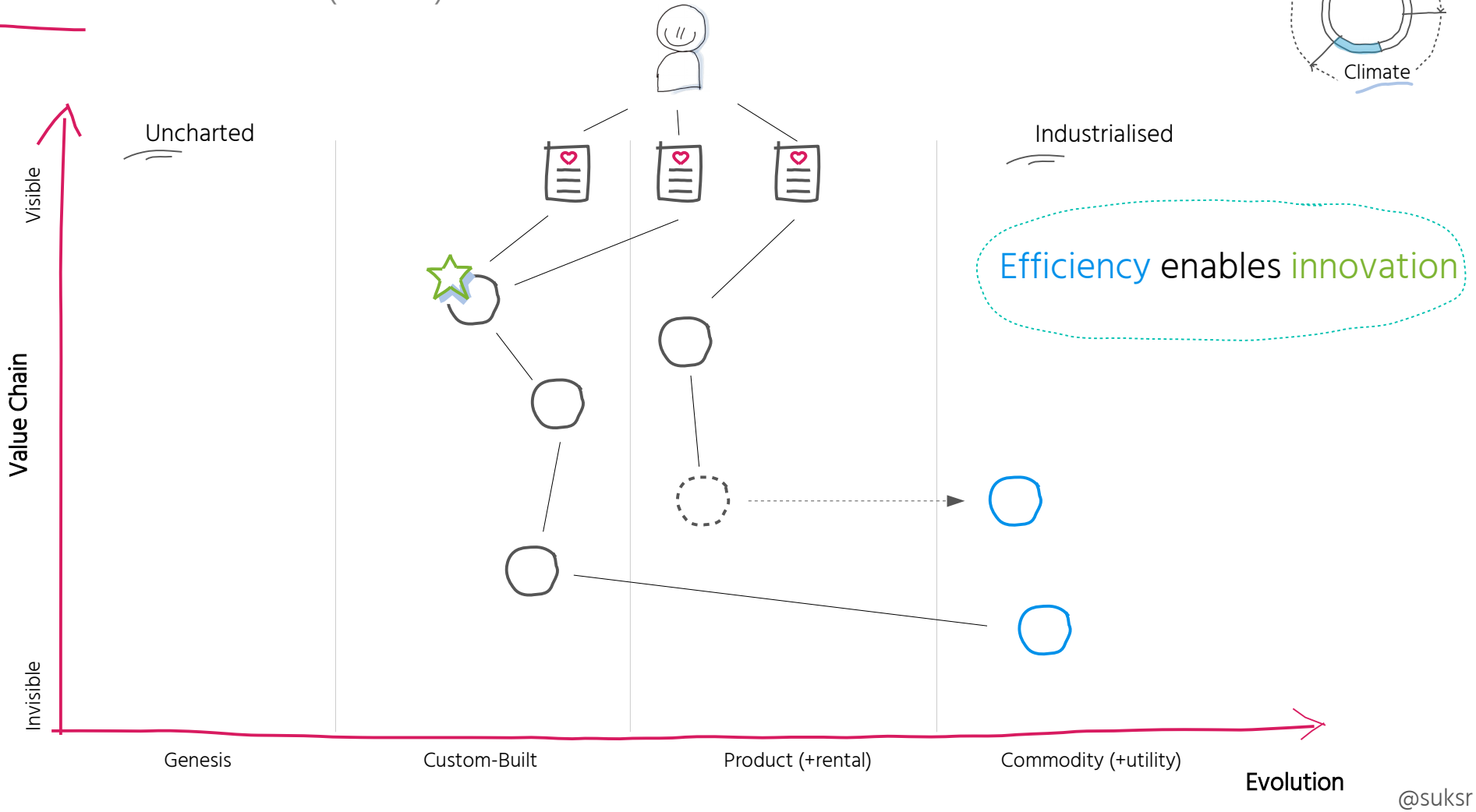




@suksr

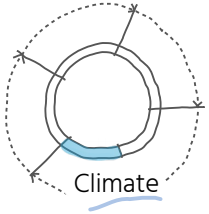
Climatic Patterns (extract)

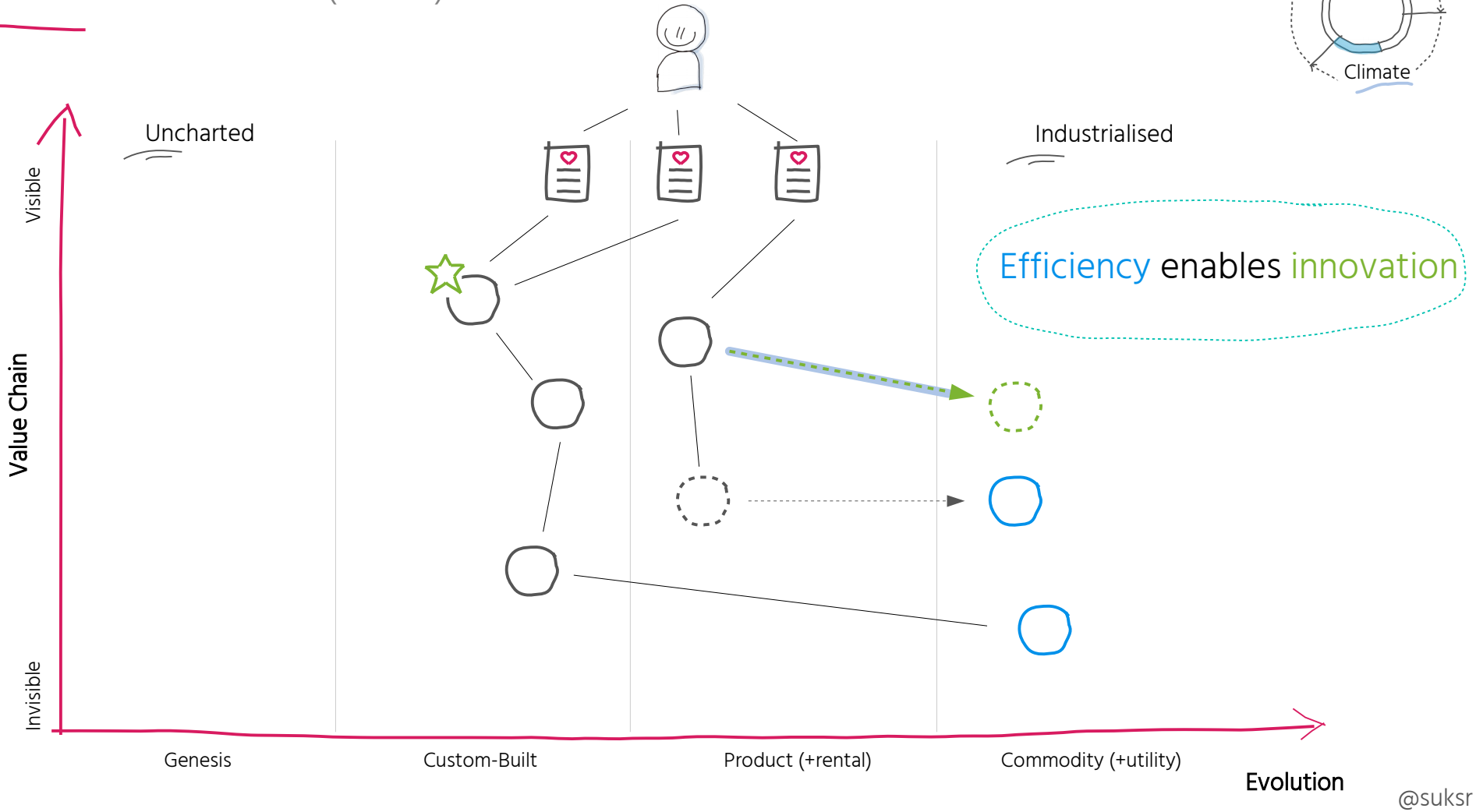




@suksr

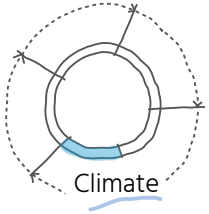
Climatic Patterns (extract)

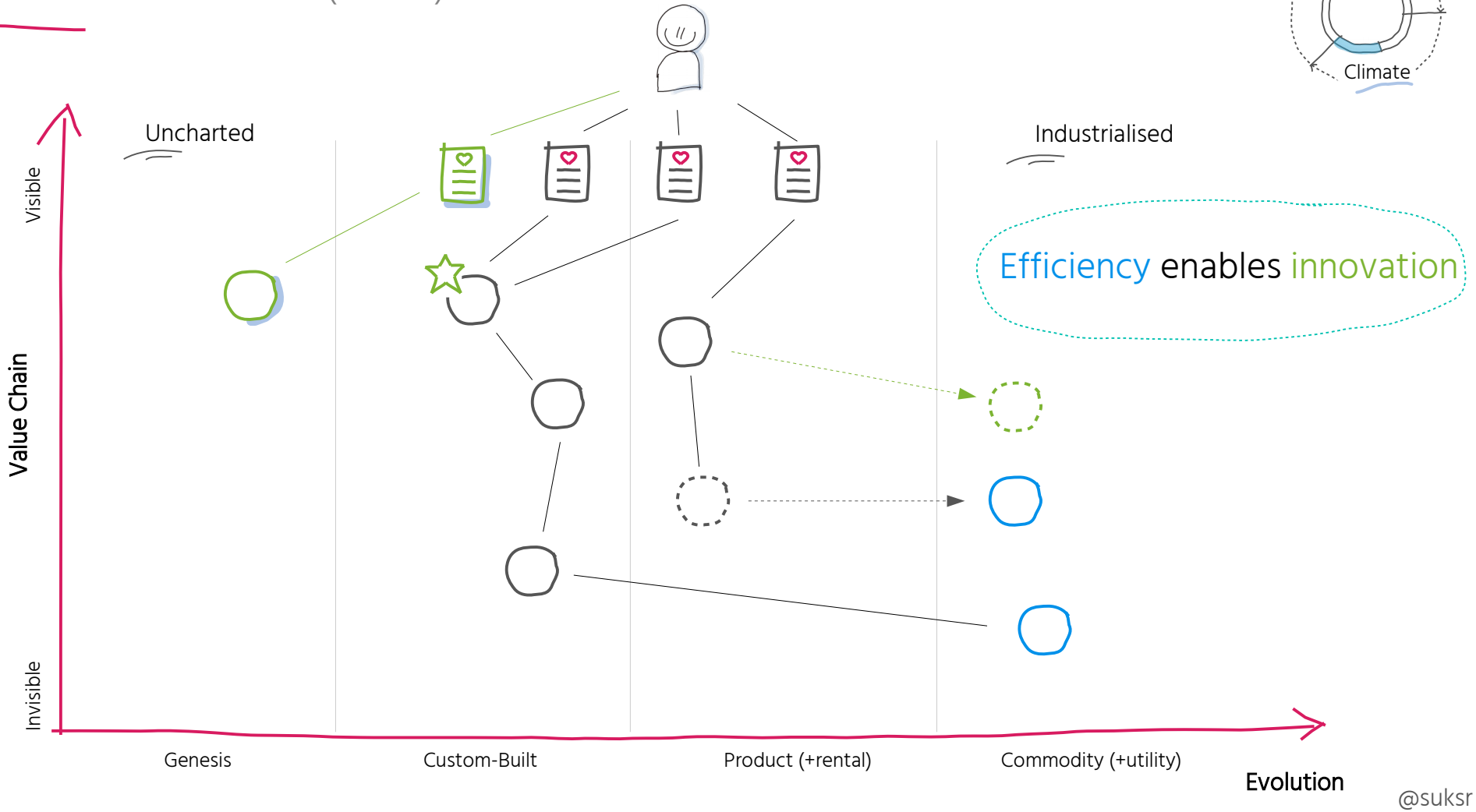




@suksr

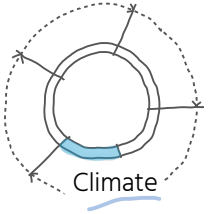
Climatic Patterns (extract)

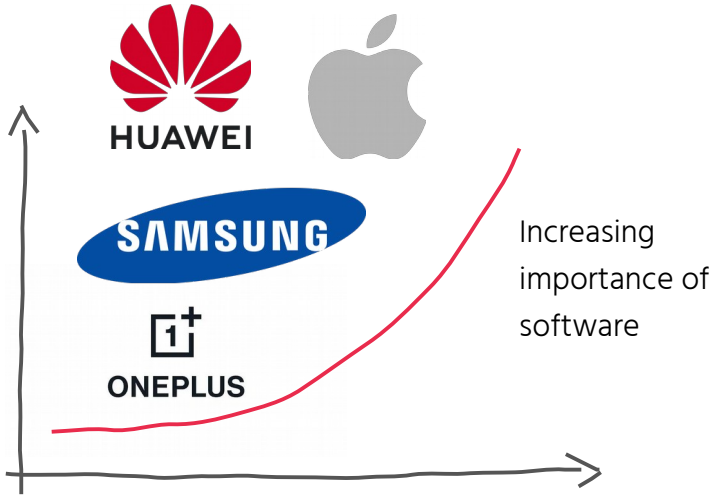




@suksr

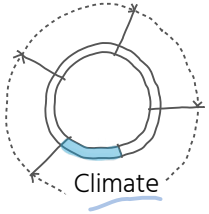
Climatic Patterns

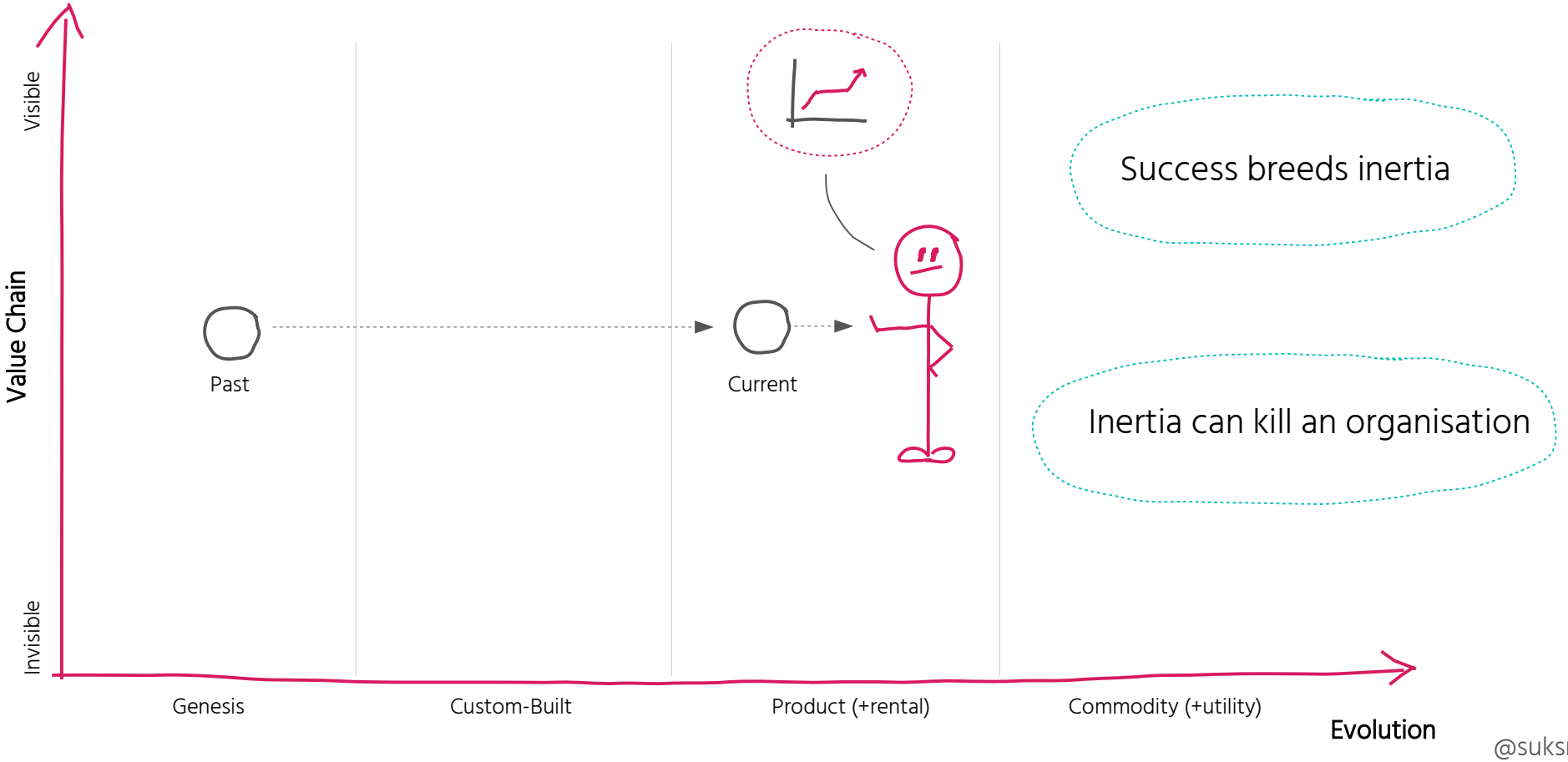




Competitors actions will change the game

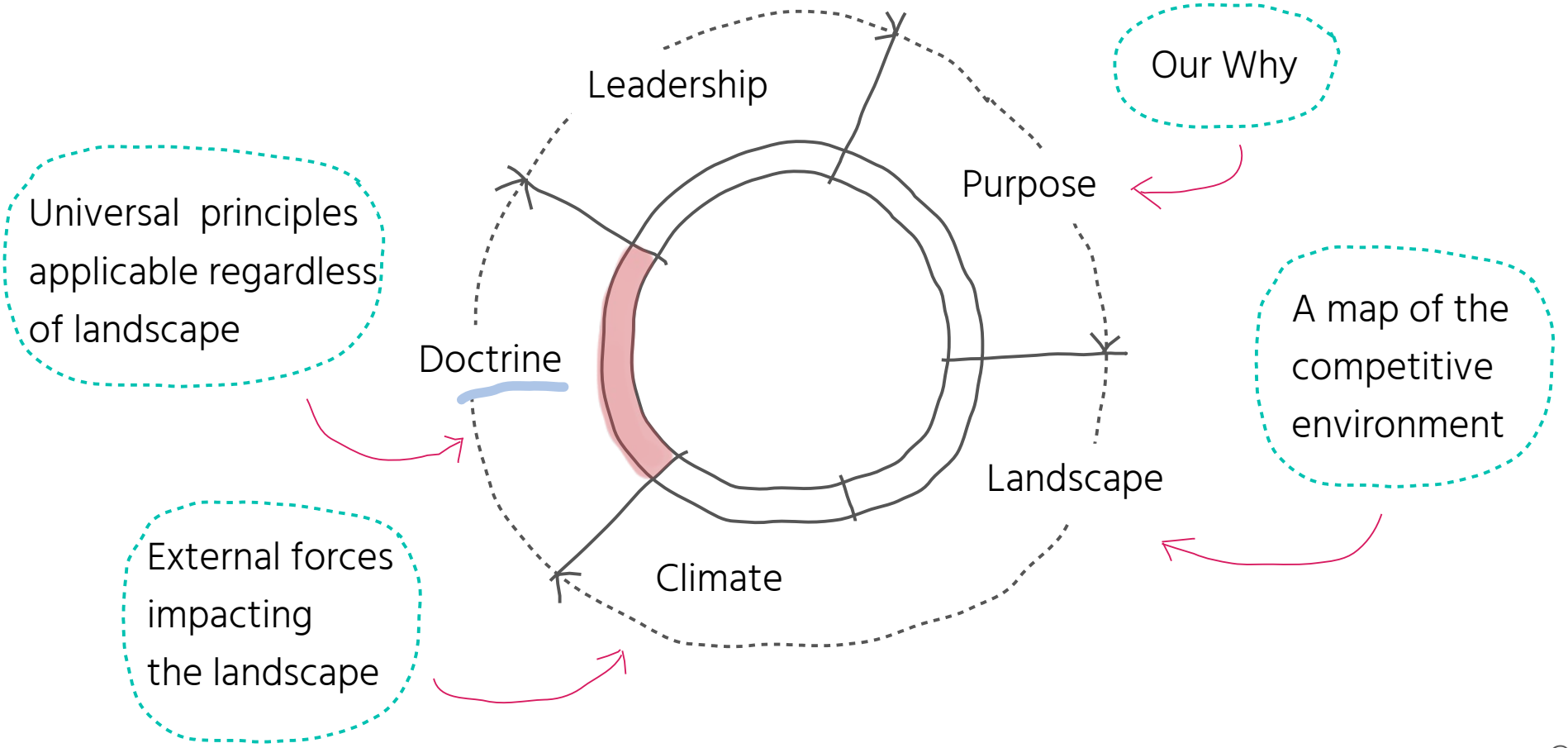
Climatic Patterns (extract)





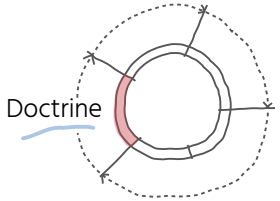
@suksr

Strategy Cycle



@suksr

Doctrinal Principles (extract)



Know your users

Value Chain

Visible

Invisible

Genesis

Custom-Built

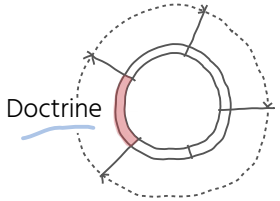
Product (+rental)

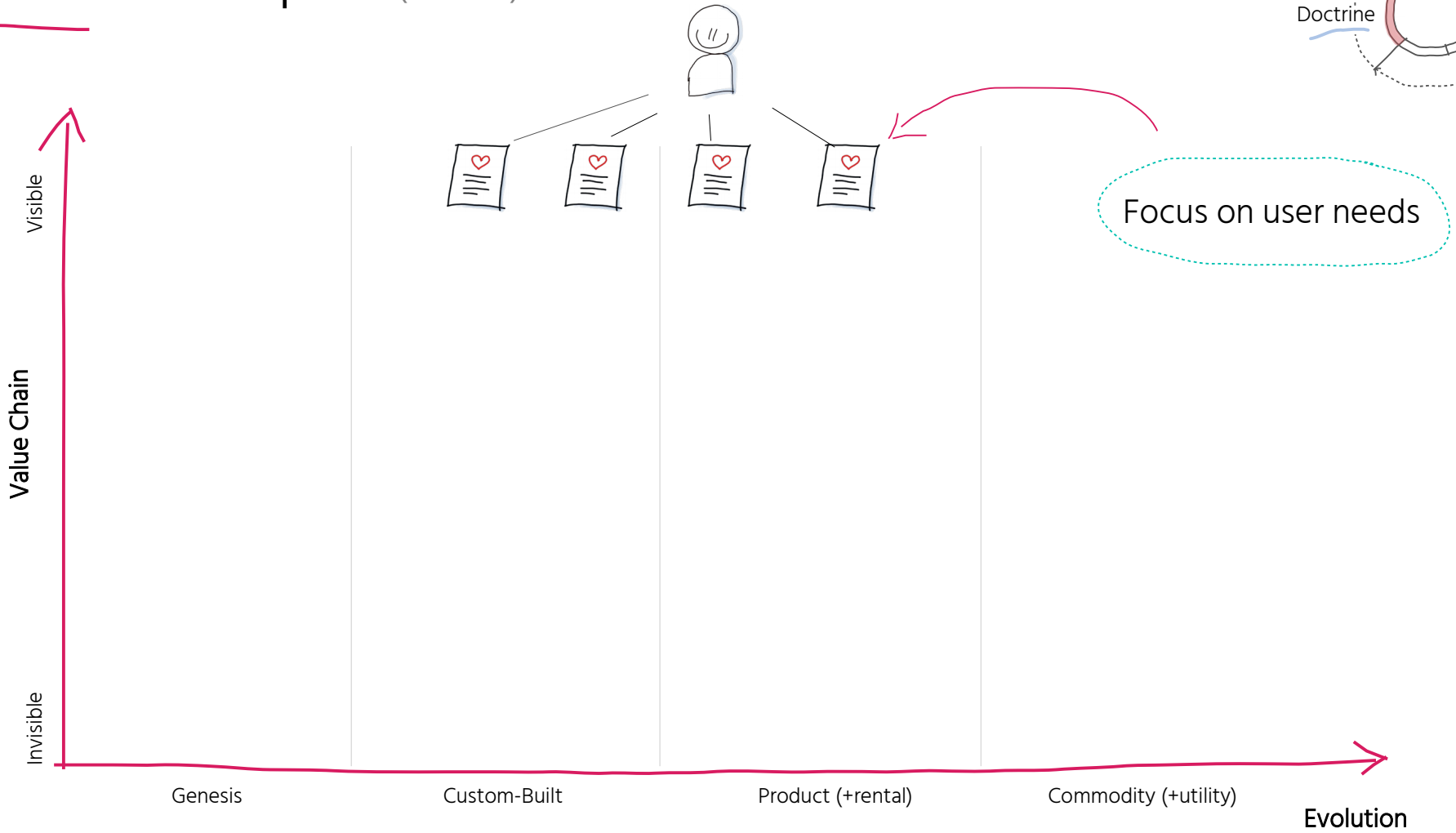
Commodity (+utility)

Evolution

@suksr

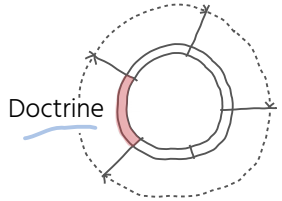
Doctrinal Principles (extract)

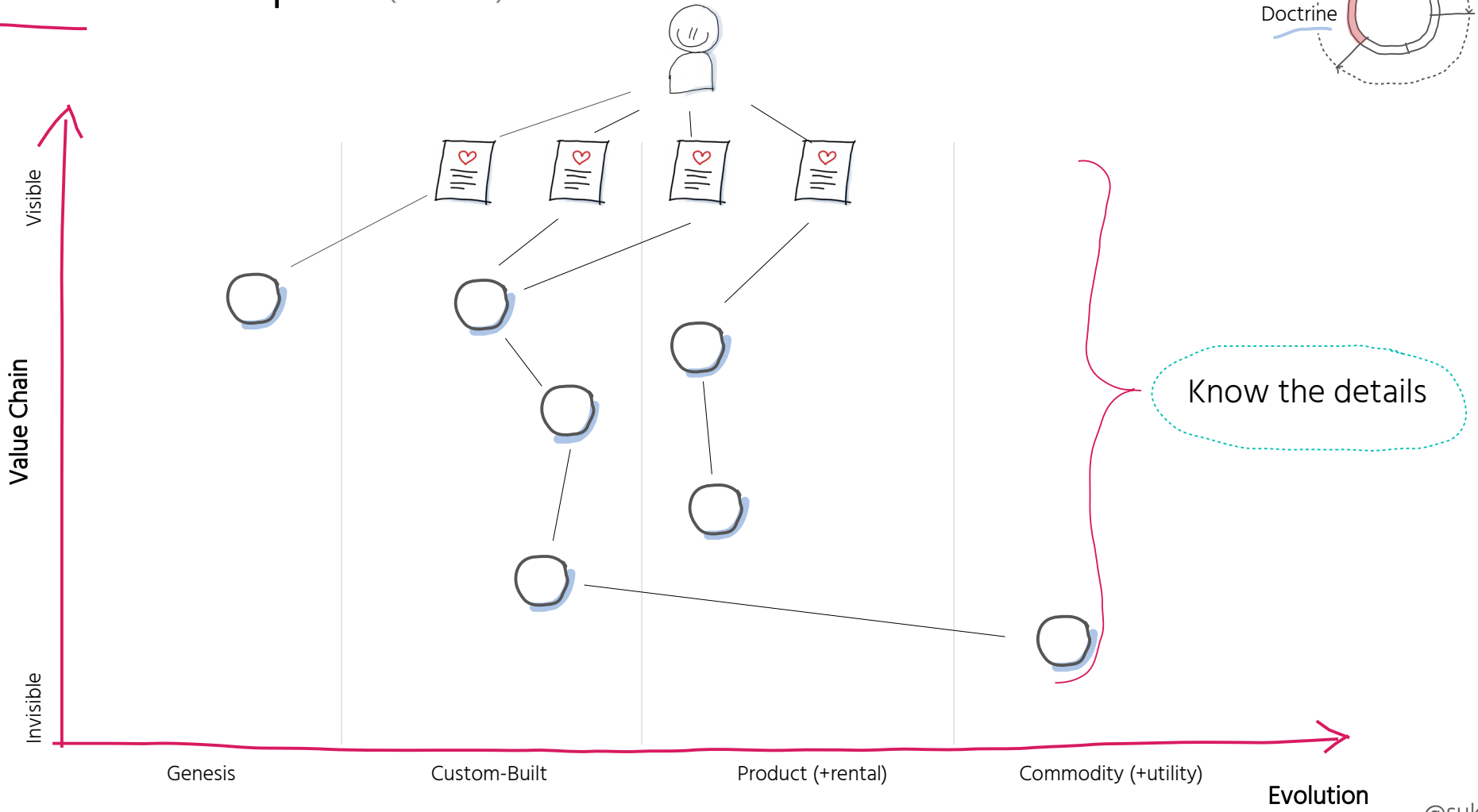




@suksr

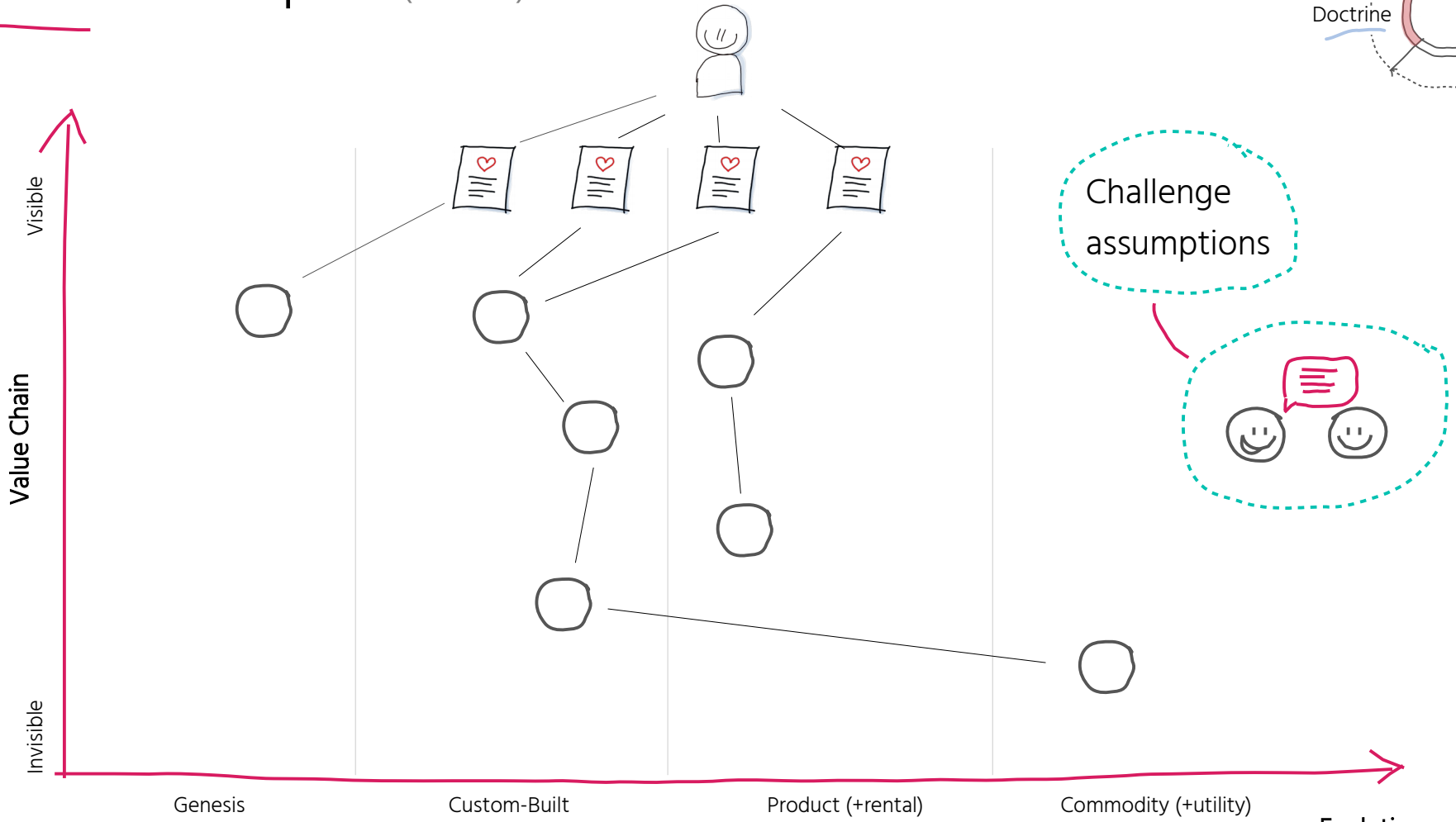
Doctrinal Principles (extract)

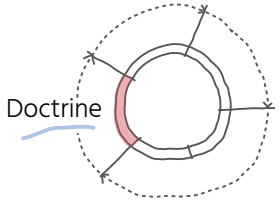




@suksr

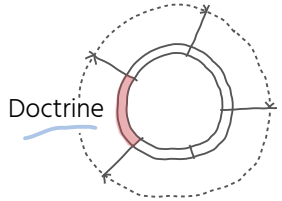
Doctrinal Principles (extract)

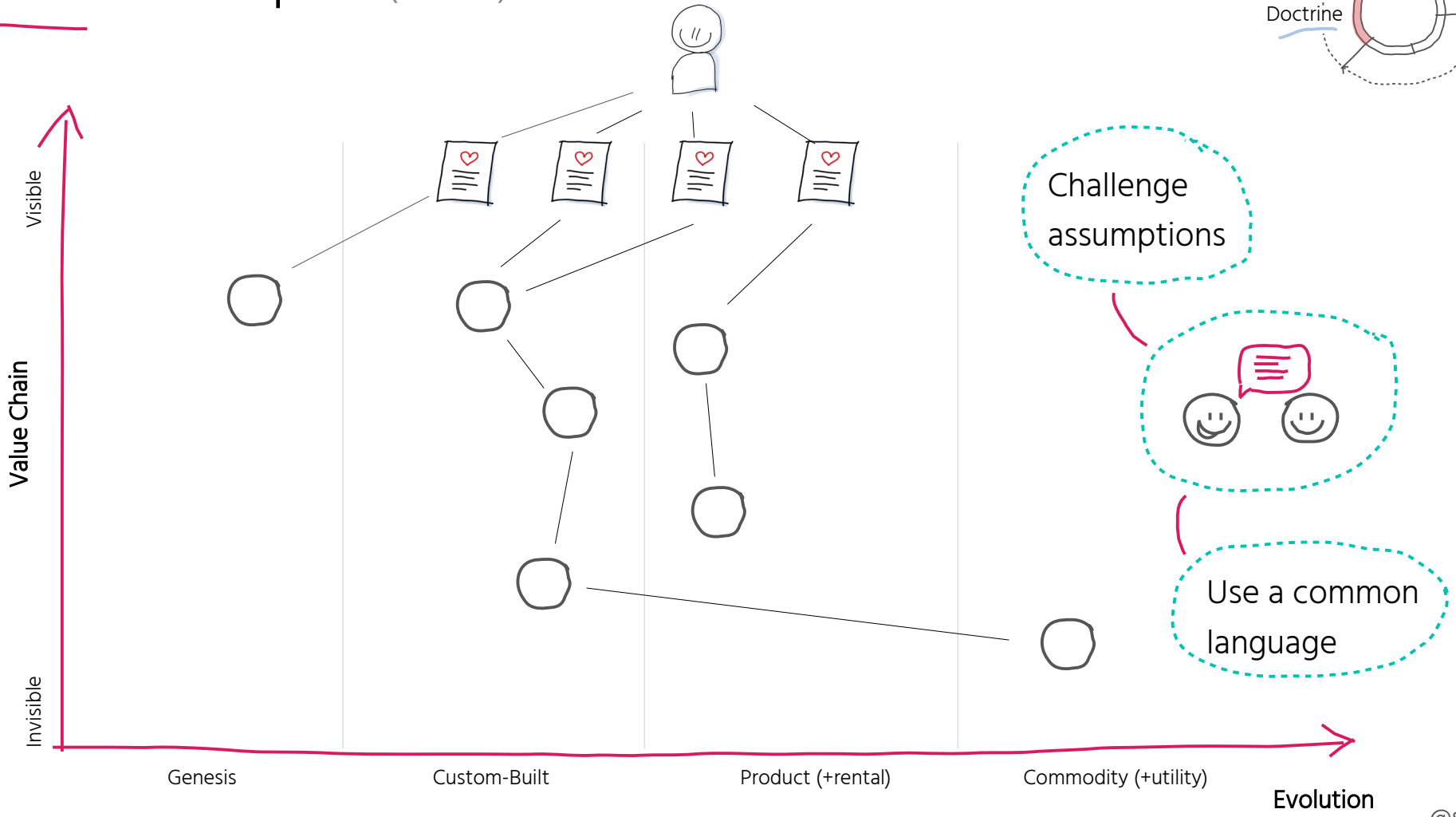




@suksr

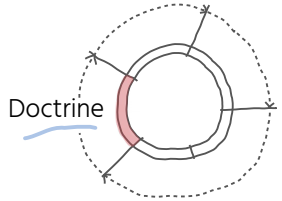
Doctrinal Principles (extract)





@suksr

Doctrinal Principles (extract)



Use appropriate methods

Build in-house

Agile

Value Chain

Visible

Invisible

Genesis

Custom-Built

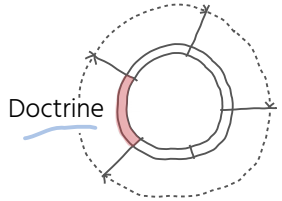
Product (+rental)

Commodity (+utility)

Evolution

@suksr

Doctrinal Principles (extract)



Use appropriate methods

Build in-house

Use/buy off-the-shelf product

Agile

Lean

Value Chain

Visible

Invisible

Genesis

Custom-Built

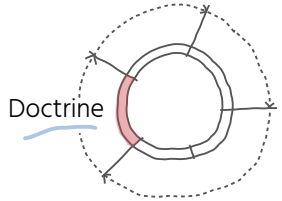
Product (+rental)

Commodity (+utility)

Evolution

@suksr

Doctrinal Principles (extract)



Use appropriate methods

Build in-house
Use/buy off-the-shelf product
Outsource to utility suppliers

Agile
Lean
Six Sigma

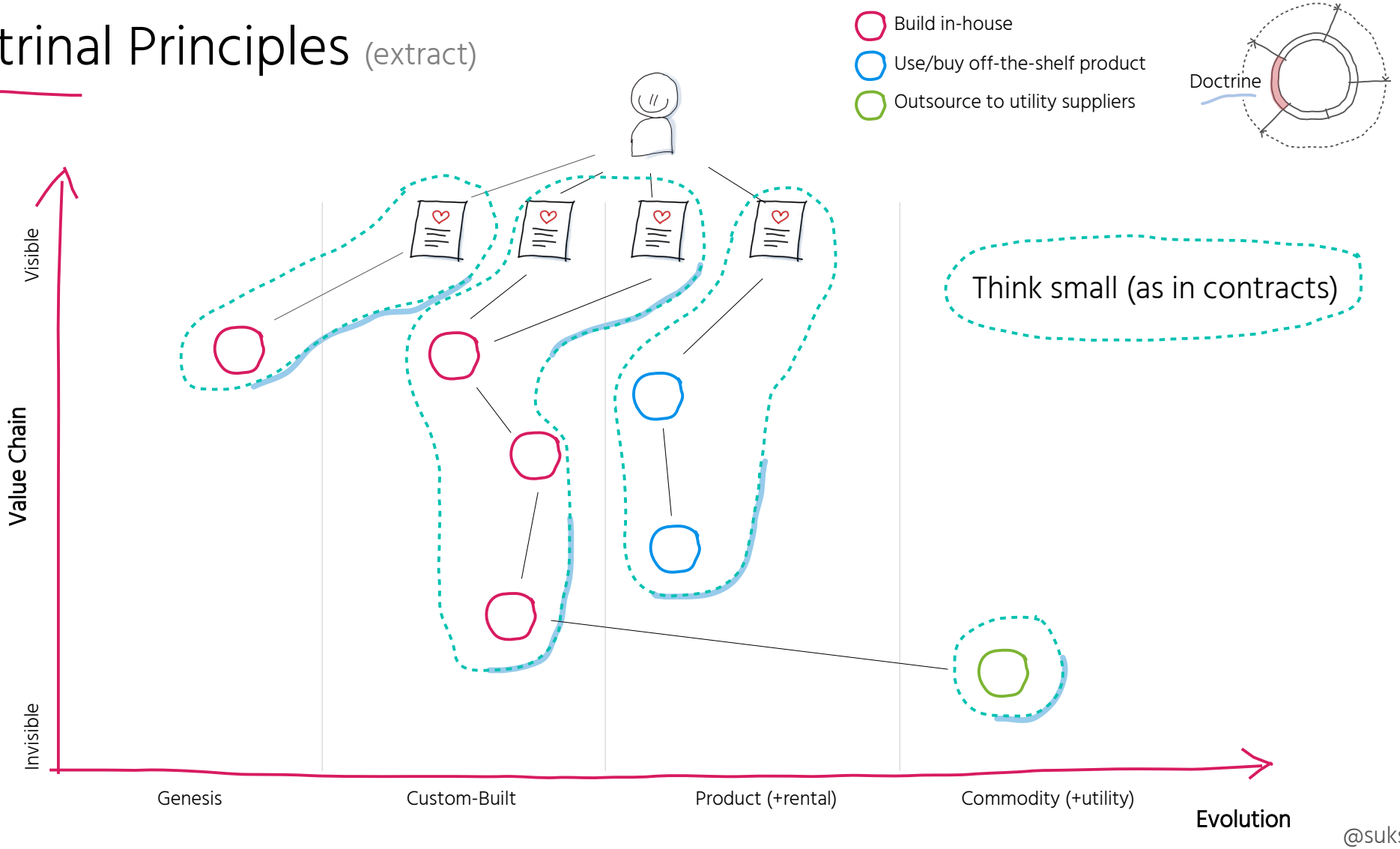
Value Chain
Visible
Invisible

Genesis
Custom-Built
Product (+rental)
Commodity (+utility)

Evolution

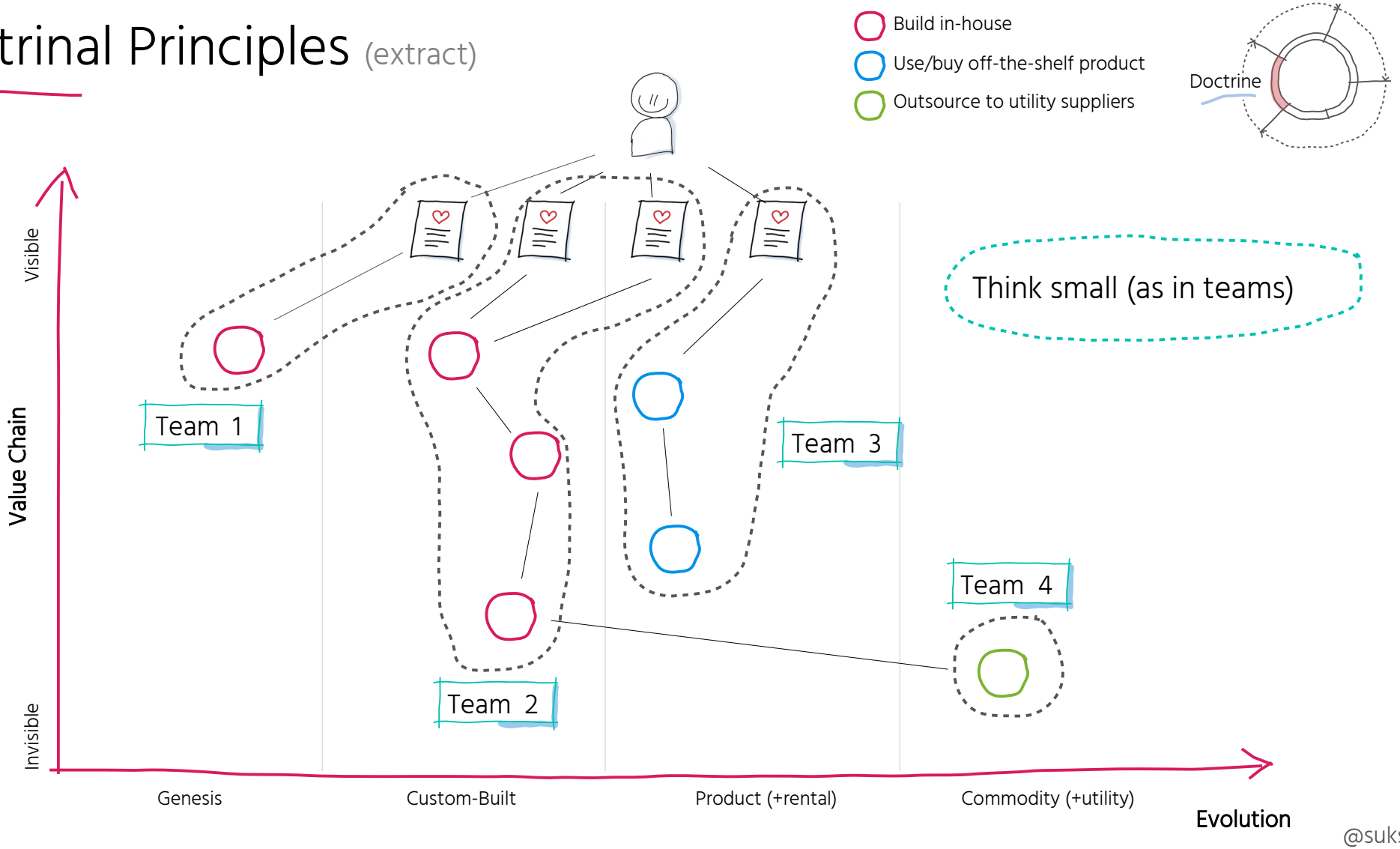
@suksr

Doctrinal Principles (extract)



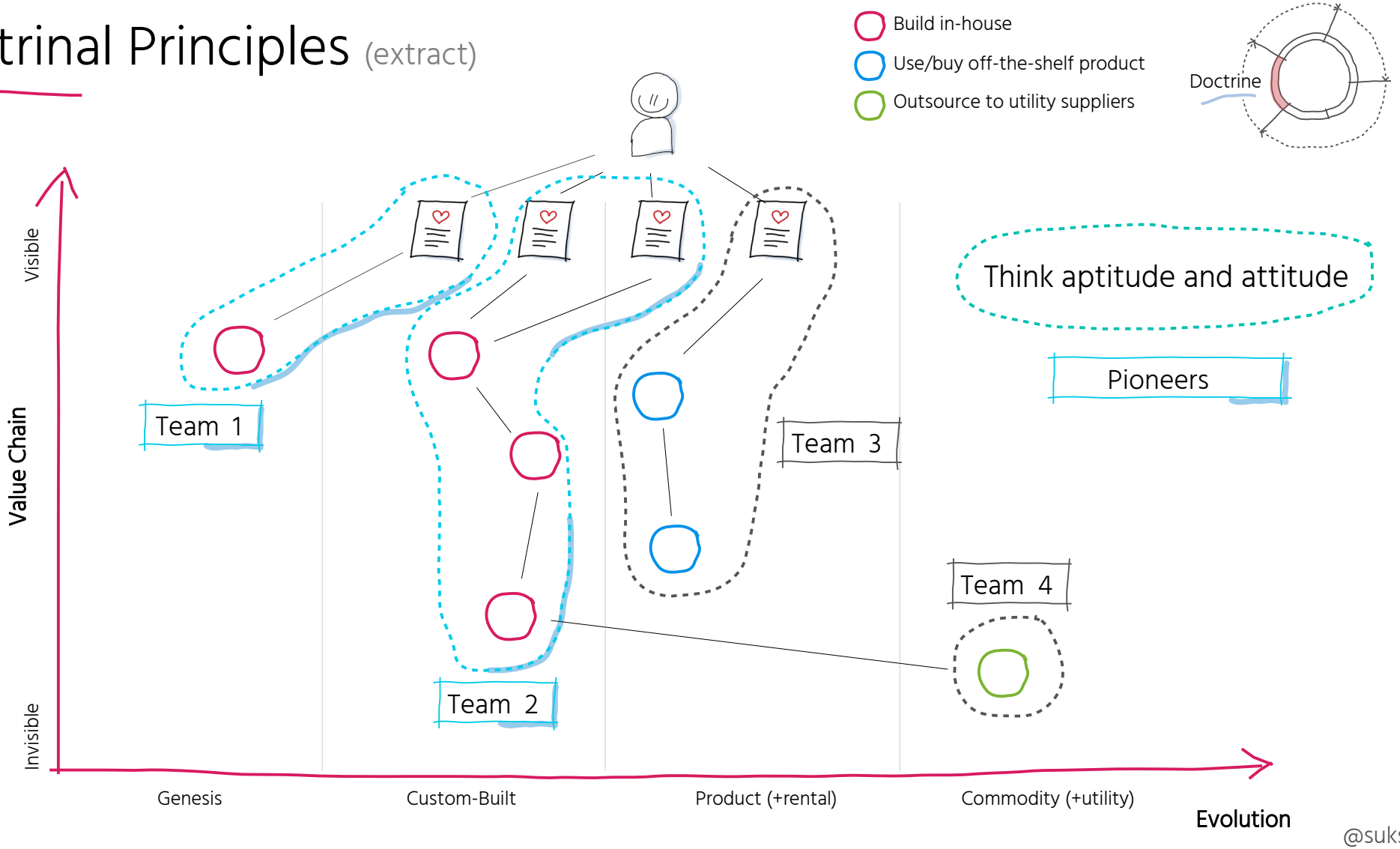
@suksr

Doctrinal Principles (extract)



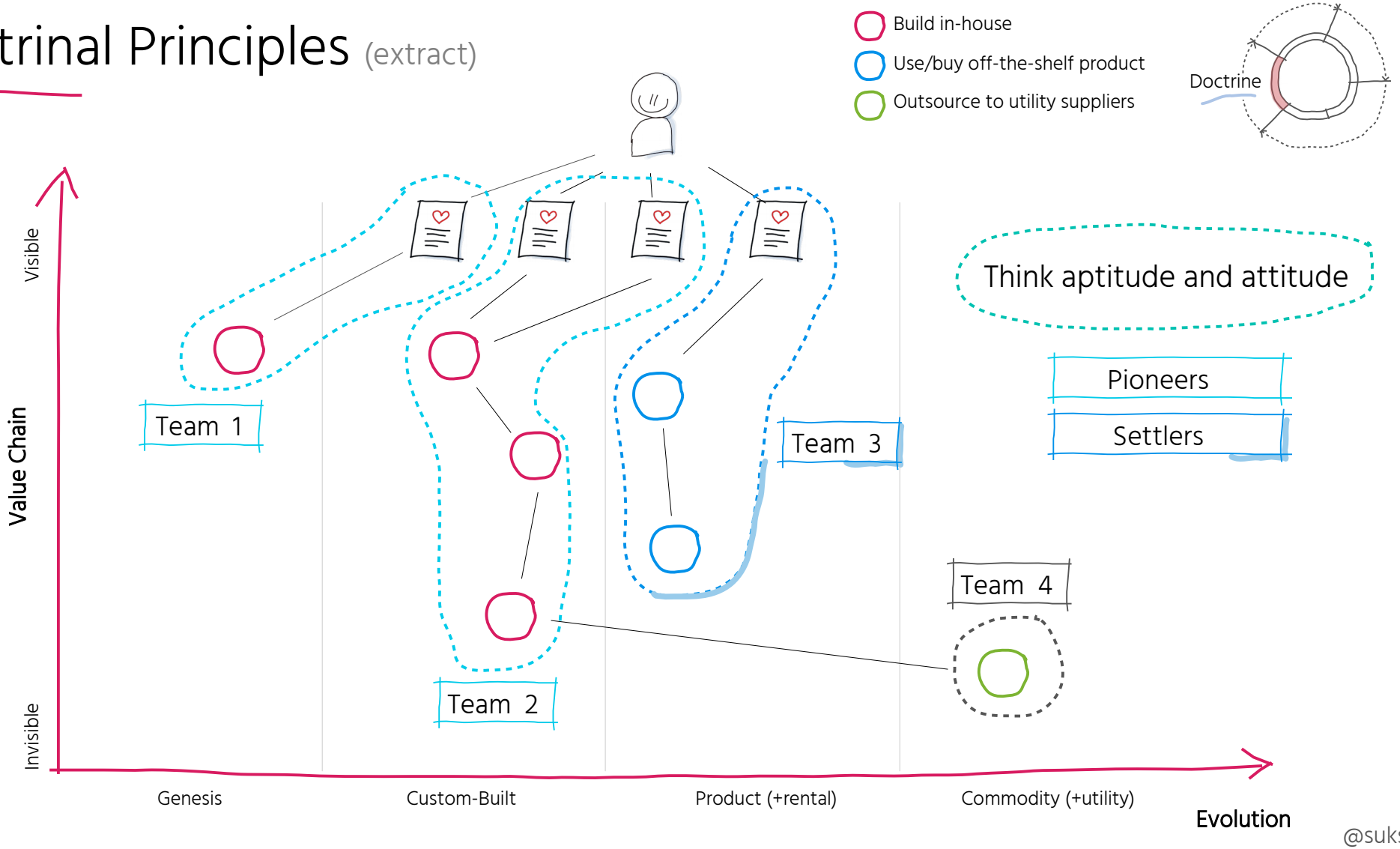
@suksr

Doctrinal Principles (extract)



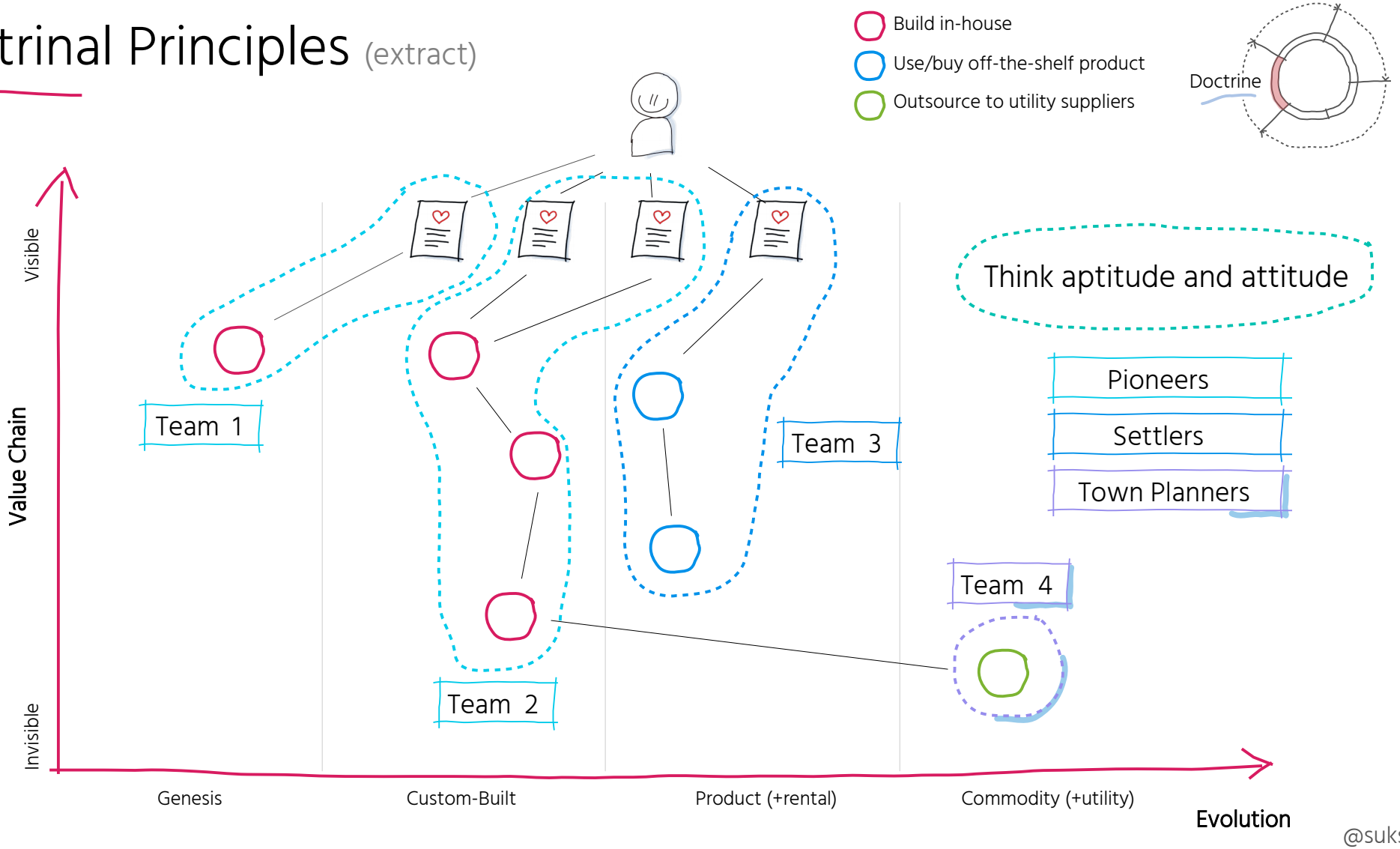
@suksr

Doctrinal Principles (extract)



@suksr

Doctrinal Principles (extract)



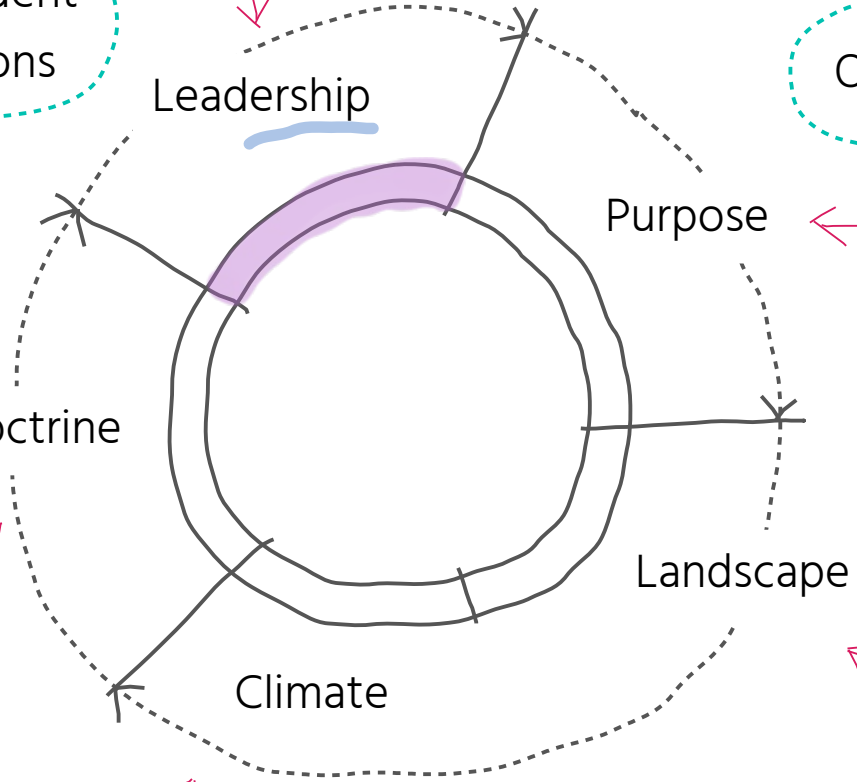
@suksr

Strategy Cycle

Context dependent strategic decisions

Universal applicable principles regardless of landscape

External forces impacting the landscape

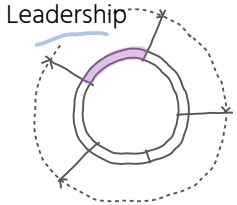


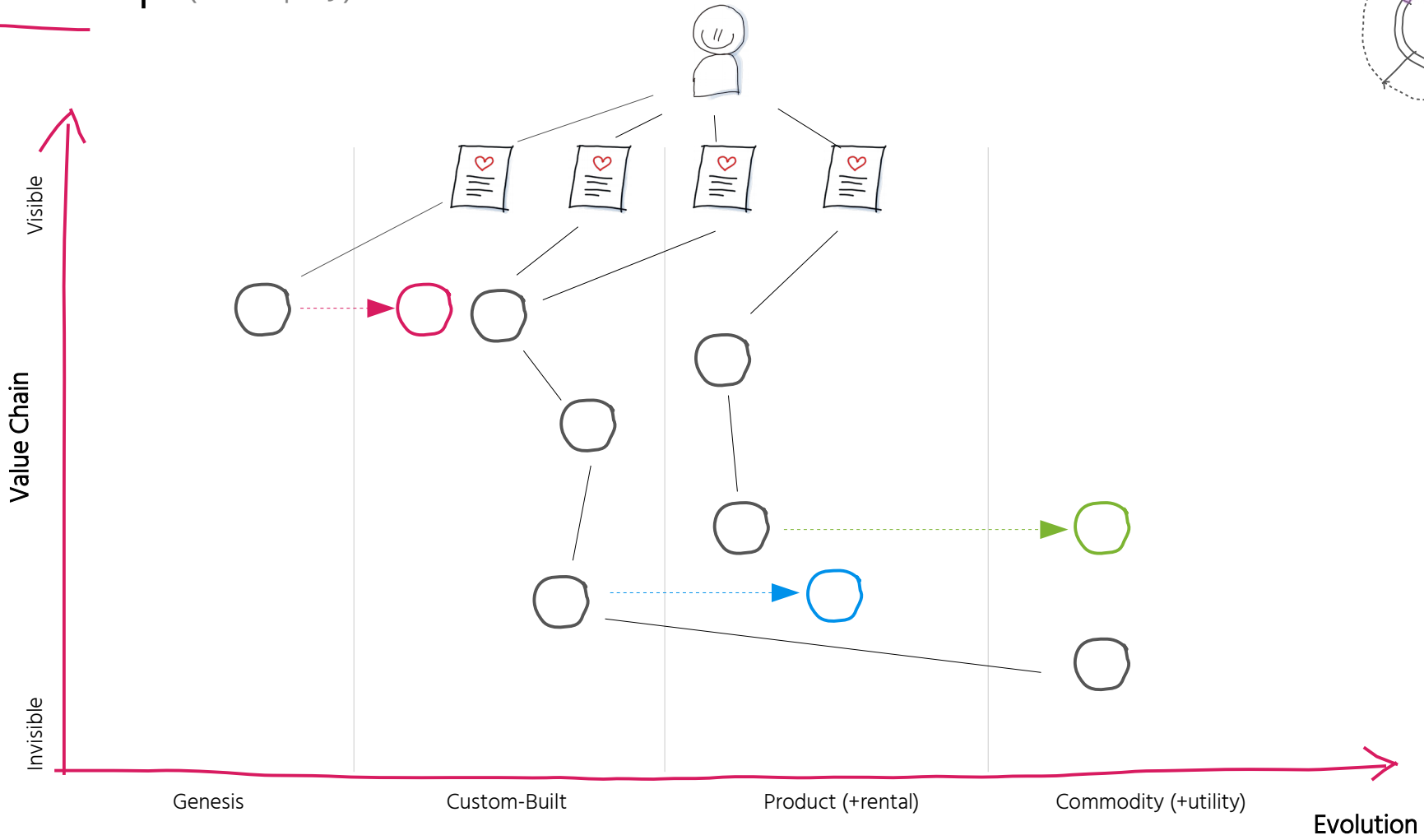
Our Why

A map of the competitive environment

@suksr

Leadership (Gameplay)



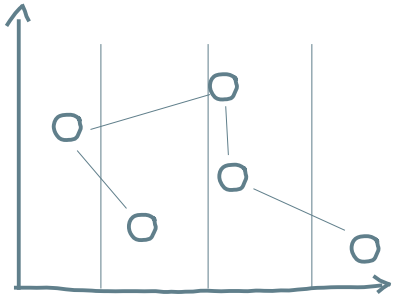


@suksr

3 Perspectives to Build Adaptive Systems

1.

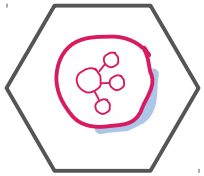
Business-Strategy



w/ Wardley Maps

2.

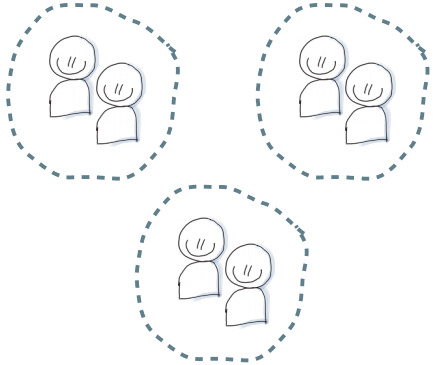
Software-Design/
-Architecture



w/ Domain-Driven Design

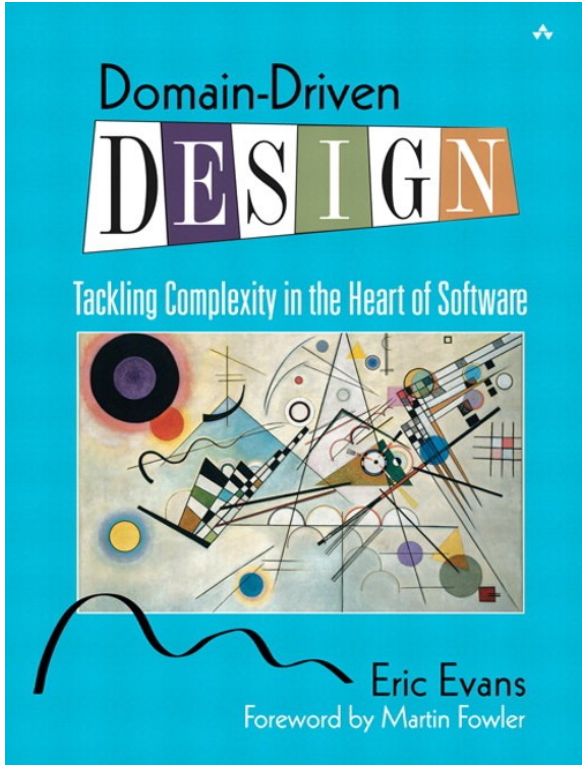
3.

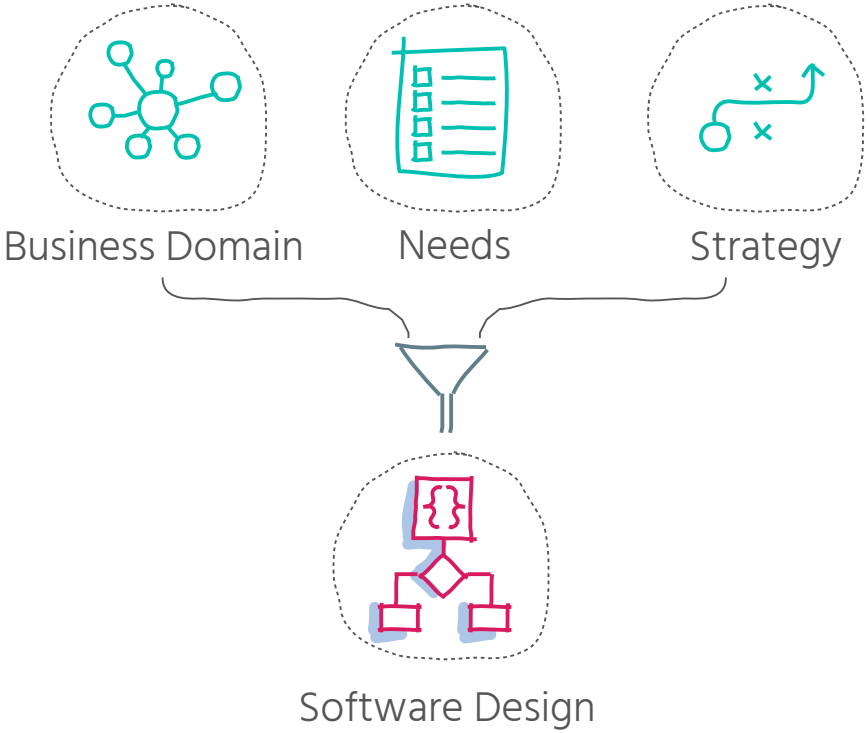
Team-Organization



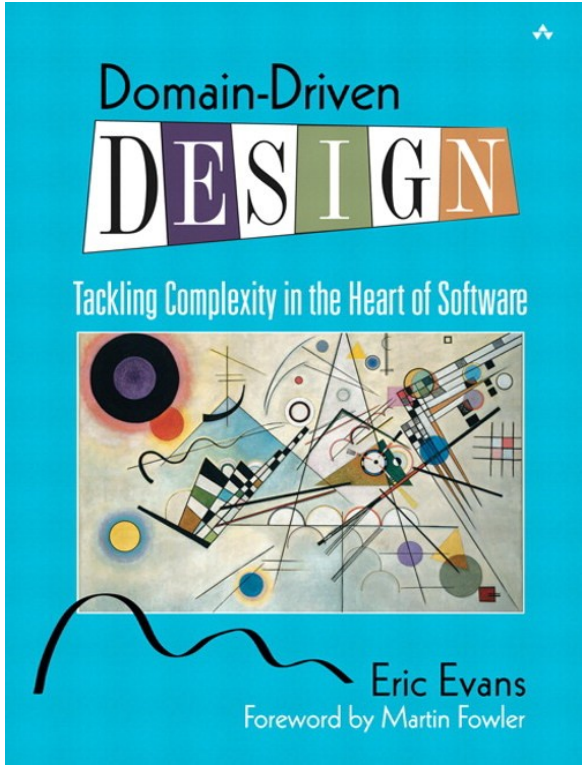
w/ Team Topologies

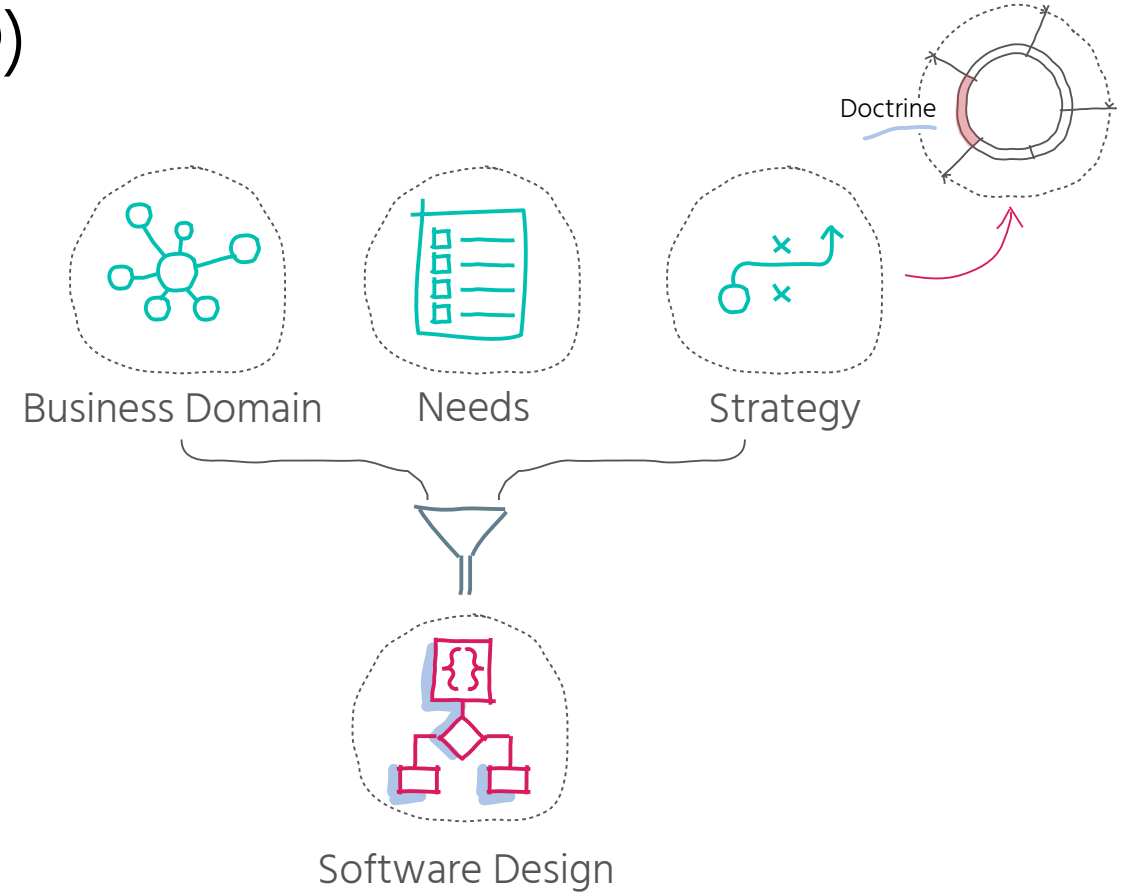
Domain Driven Design (DDD)





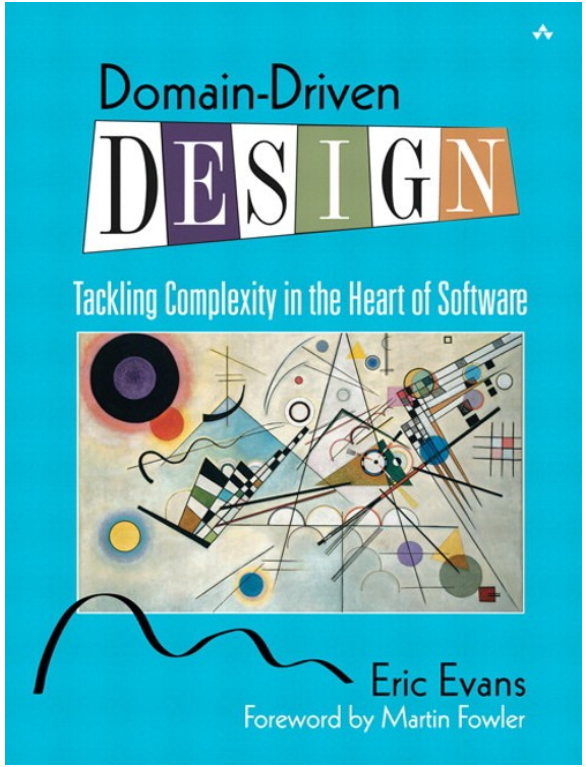
Domain Driven Design (DDD)





@suksr

Domain Driven Design (DDD) – Terminology



Strategic Design
Tactical Design

Problem Space
Solution Space

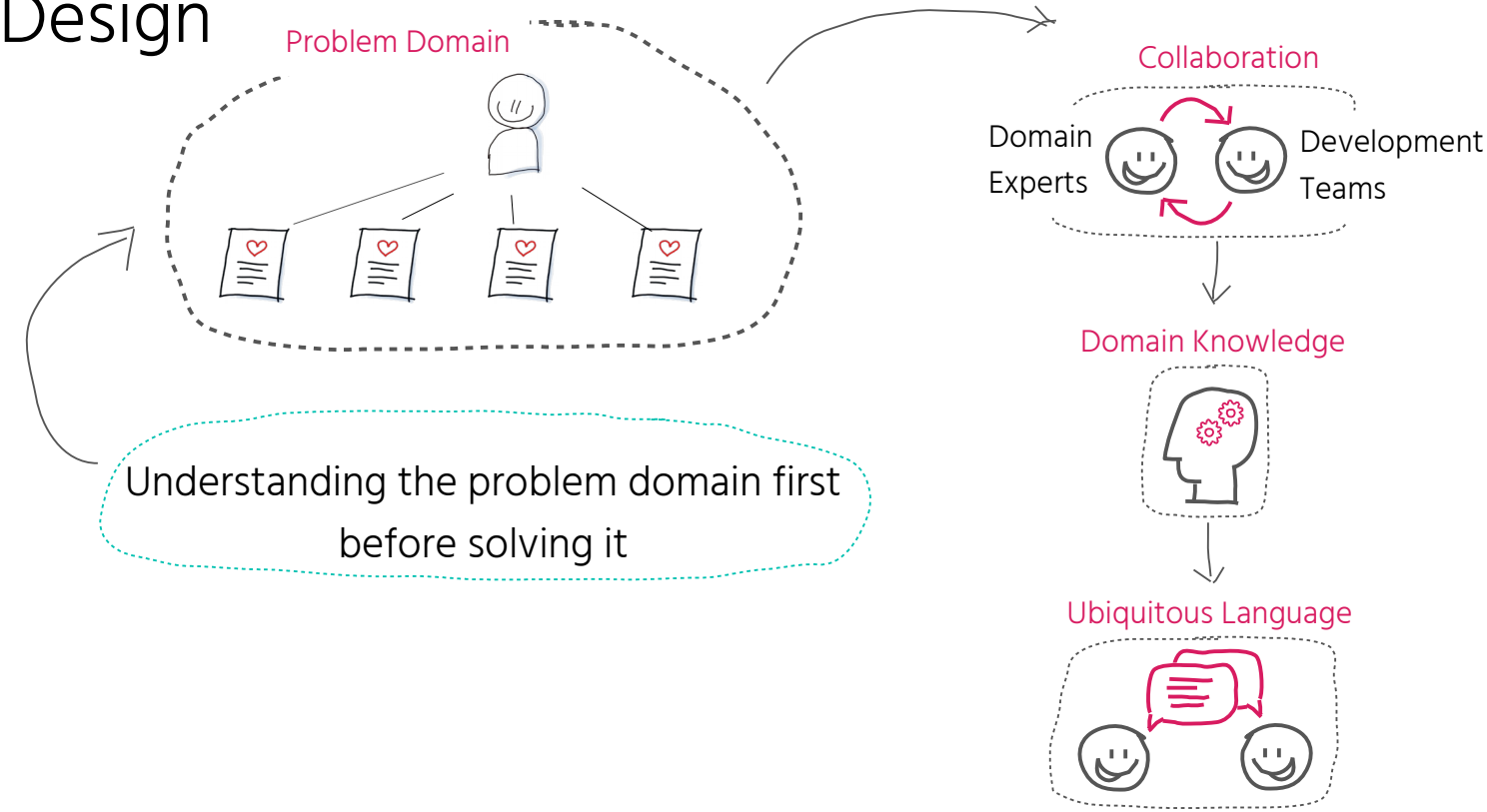
Core Subdomain
Supporting Subdomain
Generic Subdomain

Bounded Context
Ubiquitous Language

Context Maps
Anti-Corruption Layer
Shared Kernel
Open Host Service
Separate Ways
Partnership
Customer-Supplier
Conformist

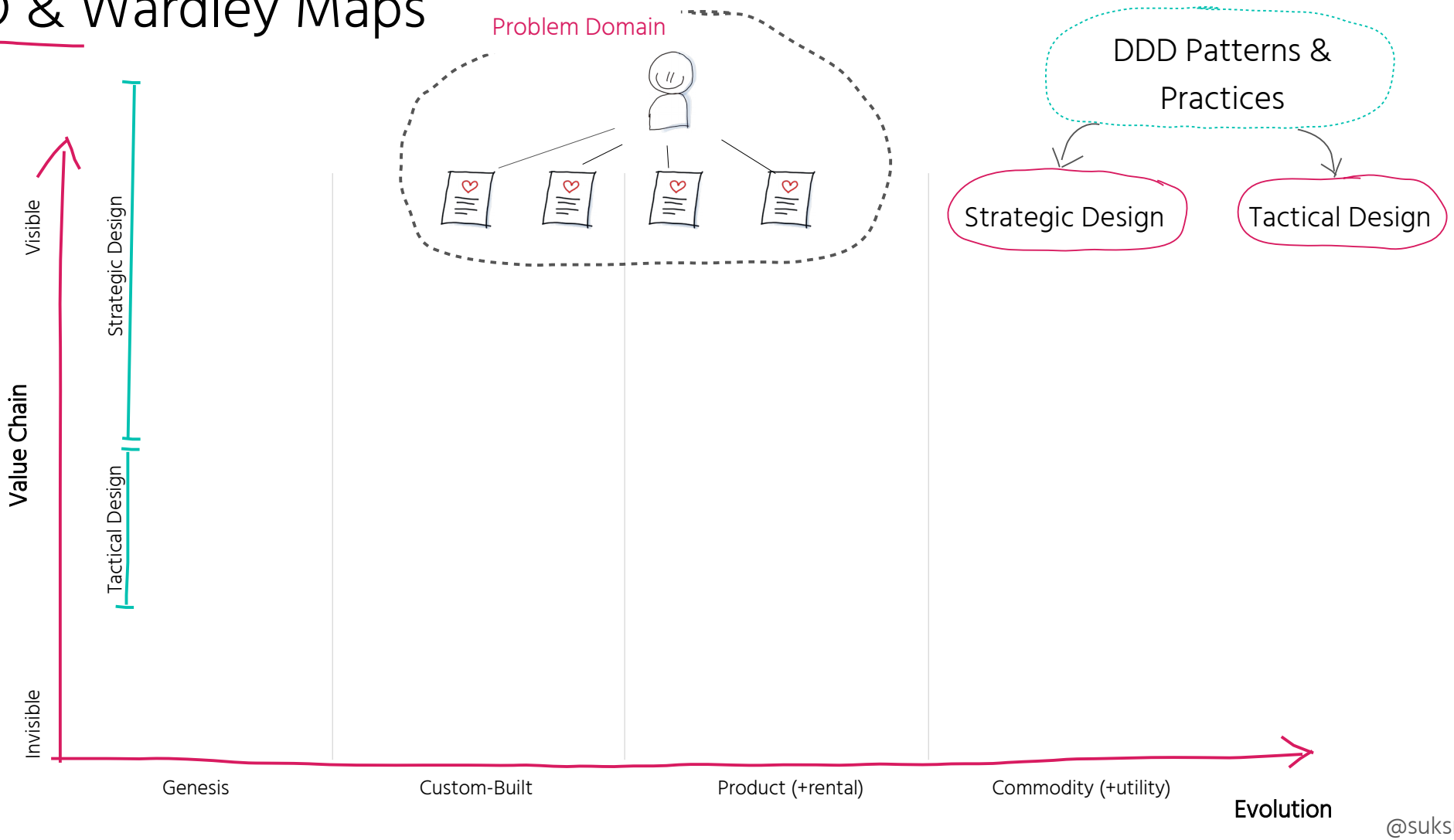
Domain Model
Entity
Value Object
Aggregate
Repository
Factory
Application Service
Domain Service
Domain Event

Domain Driven Design



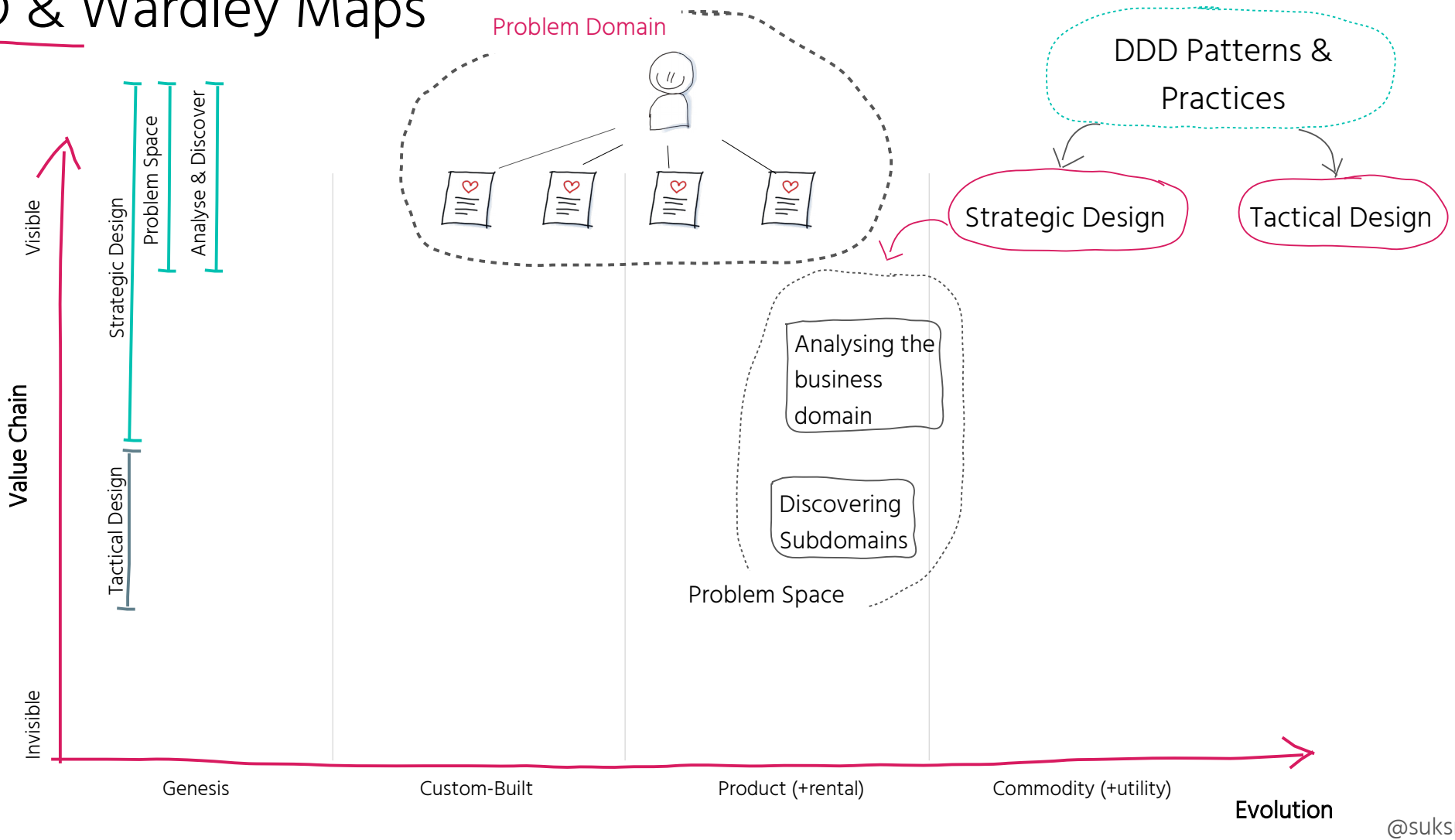
@suksr

DDD & Wardley Maps



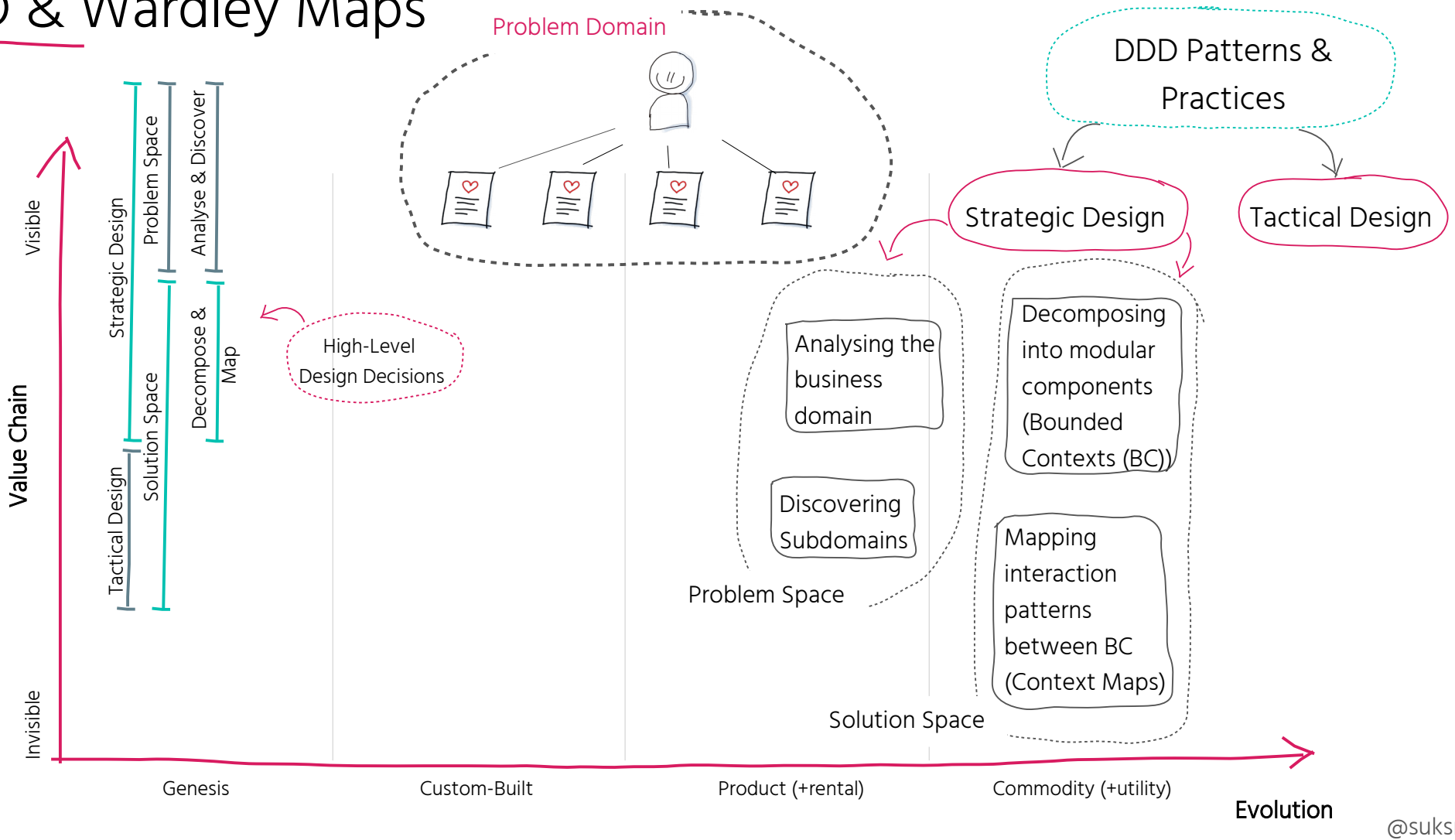
@suksr

DDD & Wardley Maps



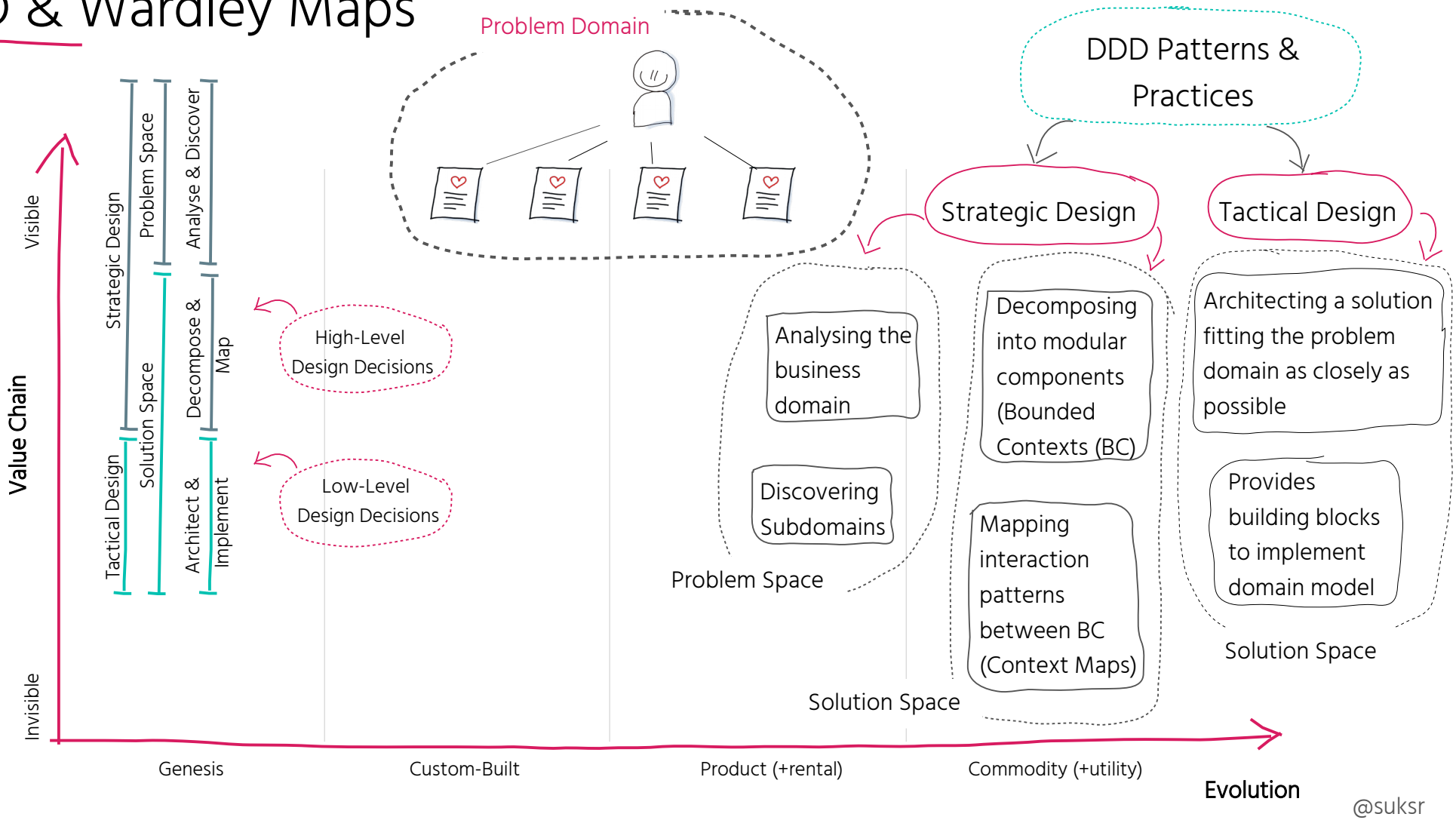
@suksr

DDD & Wardley Maps



@suksr

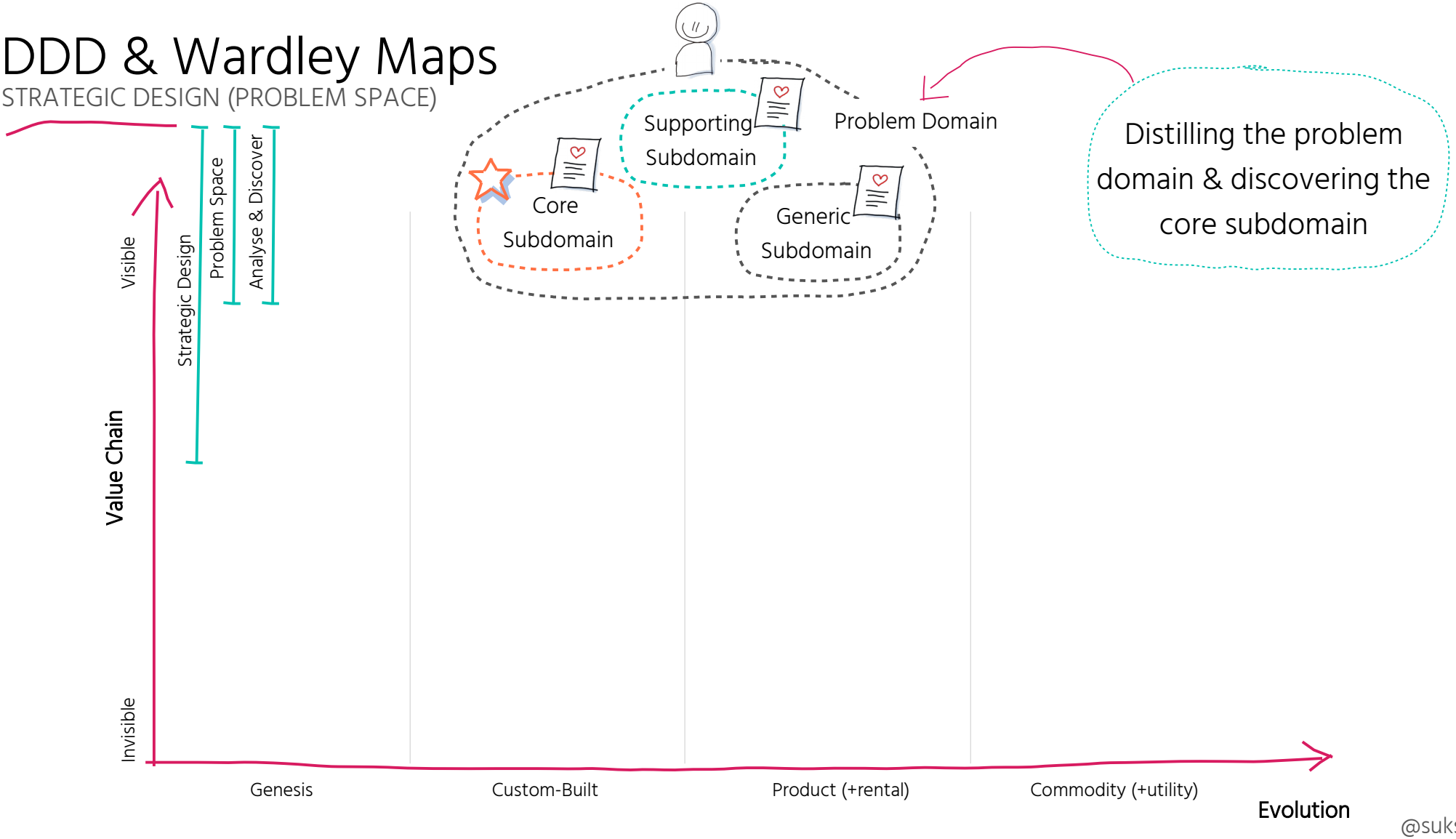
DDD & Wardley Maps



@suksr

DDD & Wardley Maps

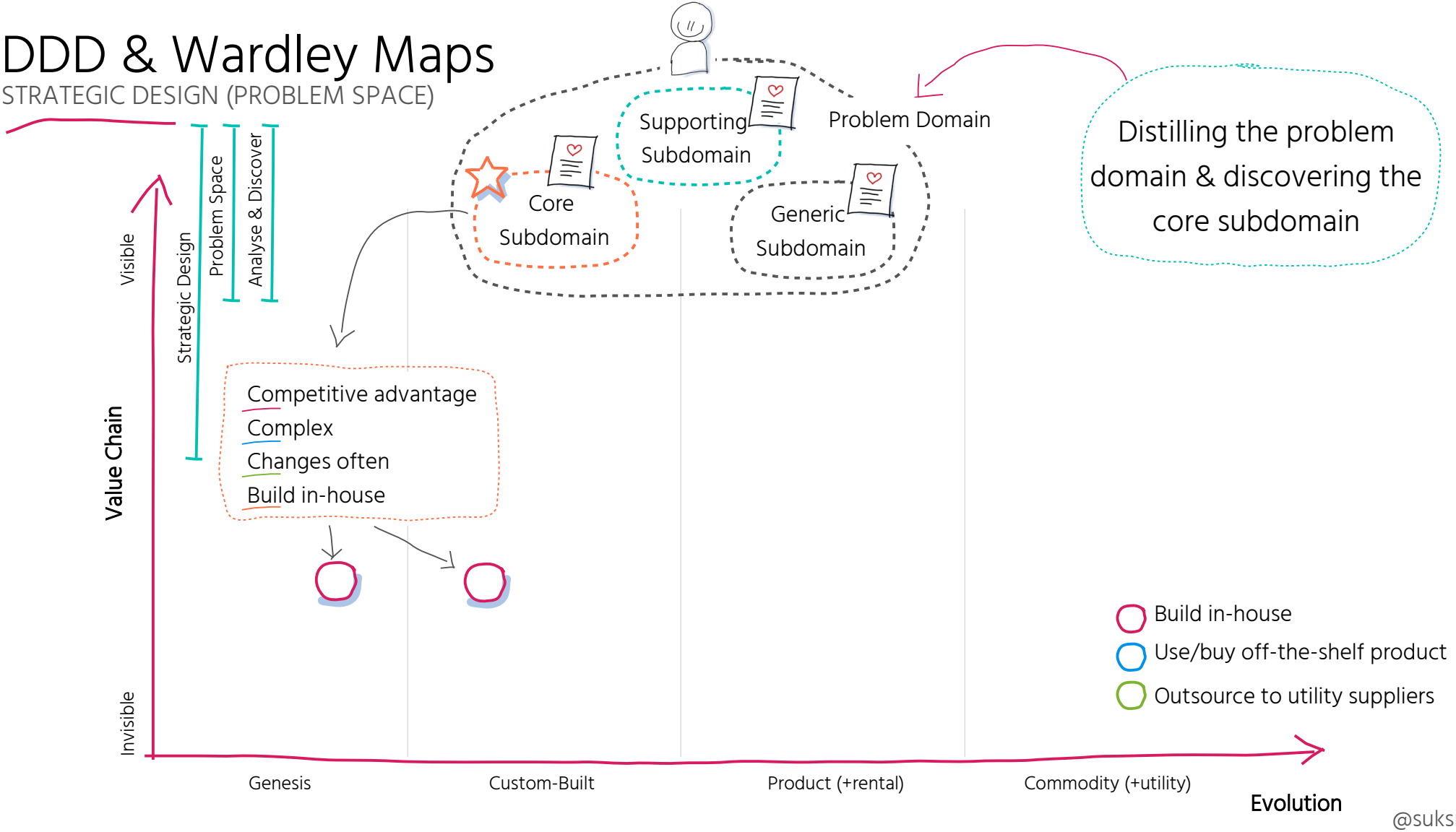
STRATEGIC DESIGN (PROBLEM SPACE)



@suksr

DDD & Wardley Maps

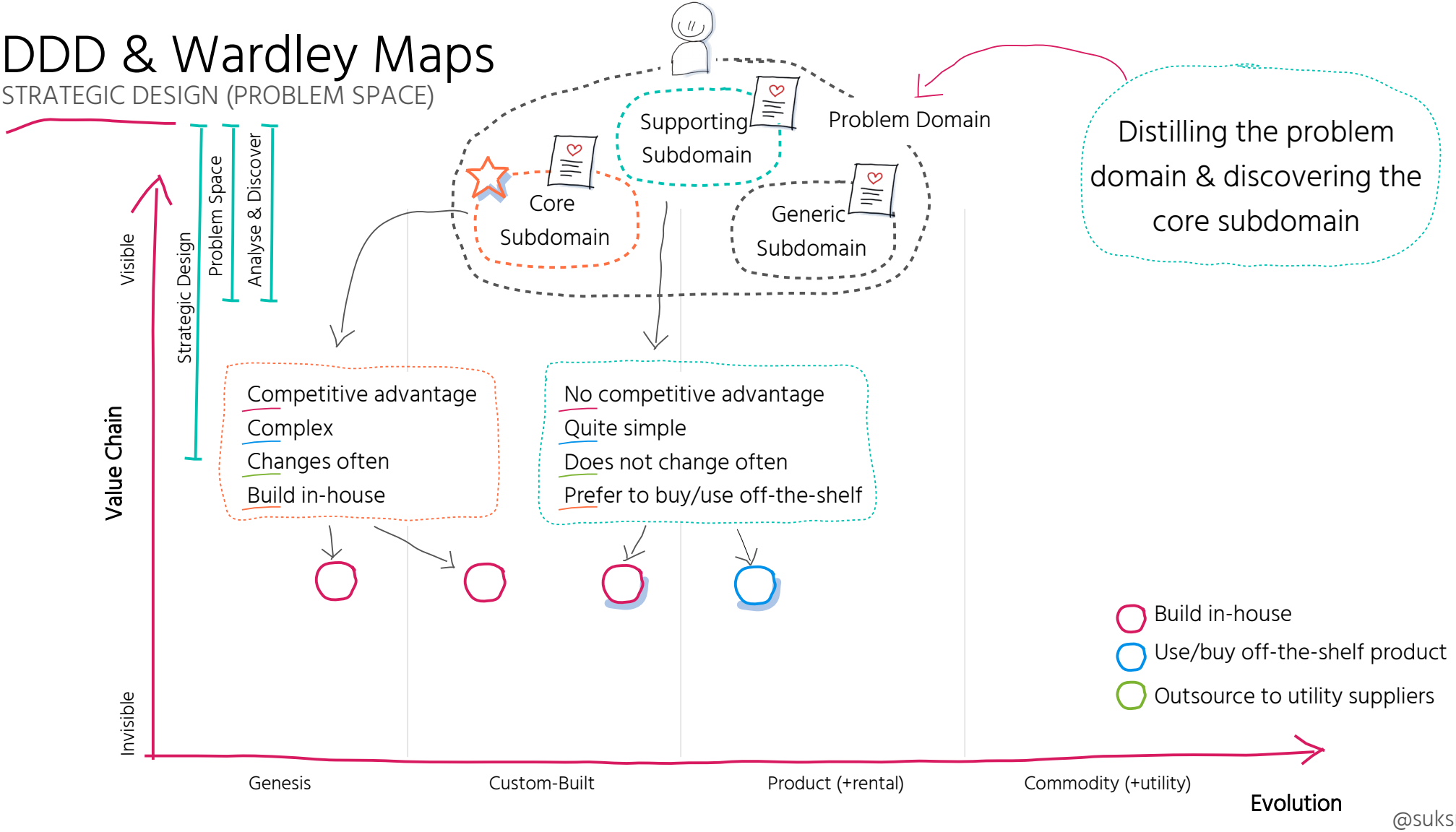
STRATEGIC DESIGN (PROBLEM SPACE)



@suksr

DDD & Wardley Maps

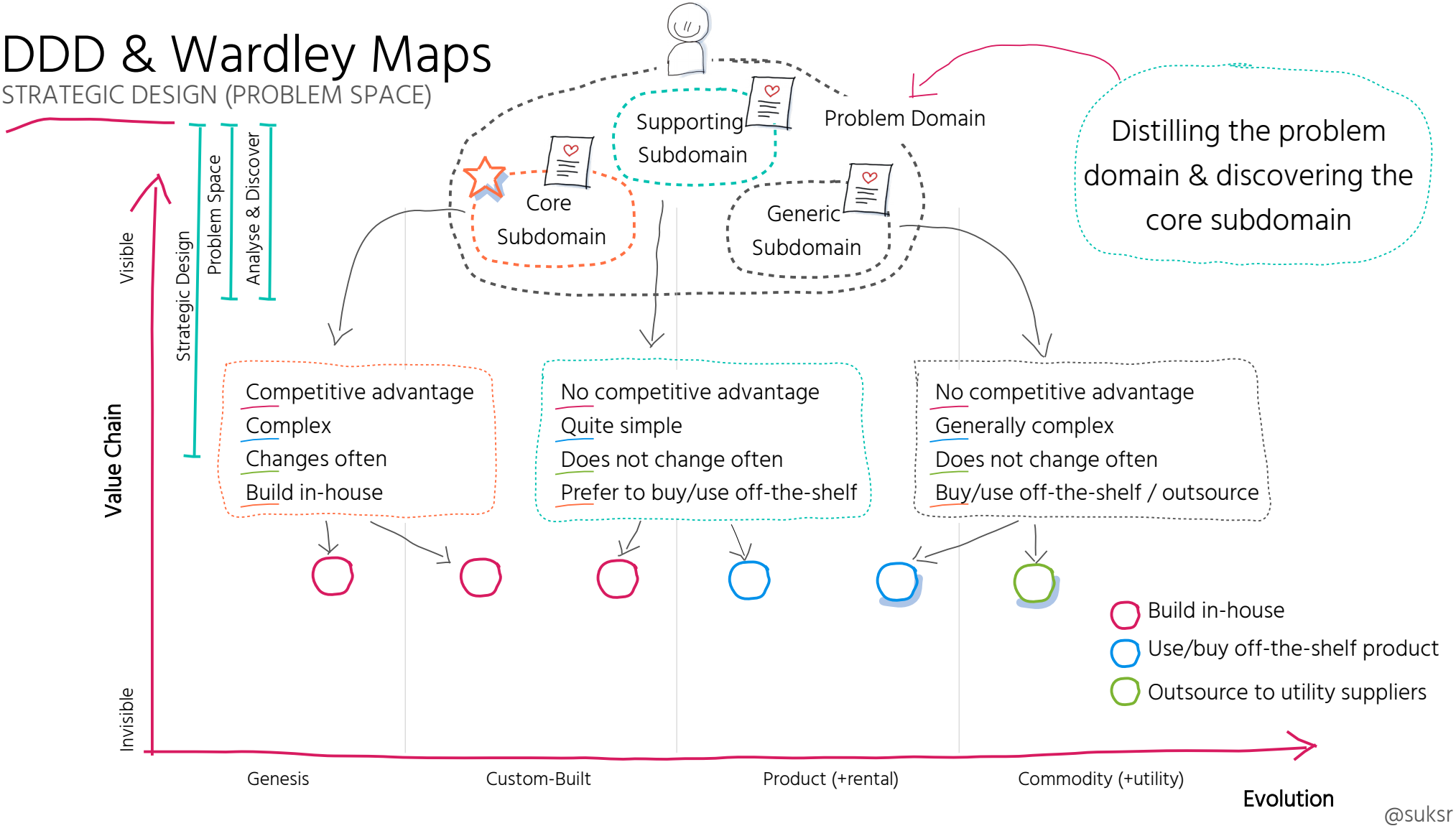
STRATEGIC DESIGN (PROBLEM SPACE)



@suksr

DDD & Wardley Maps

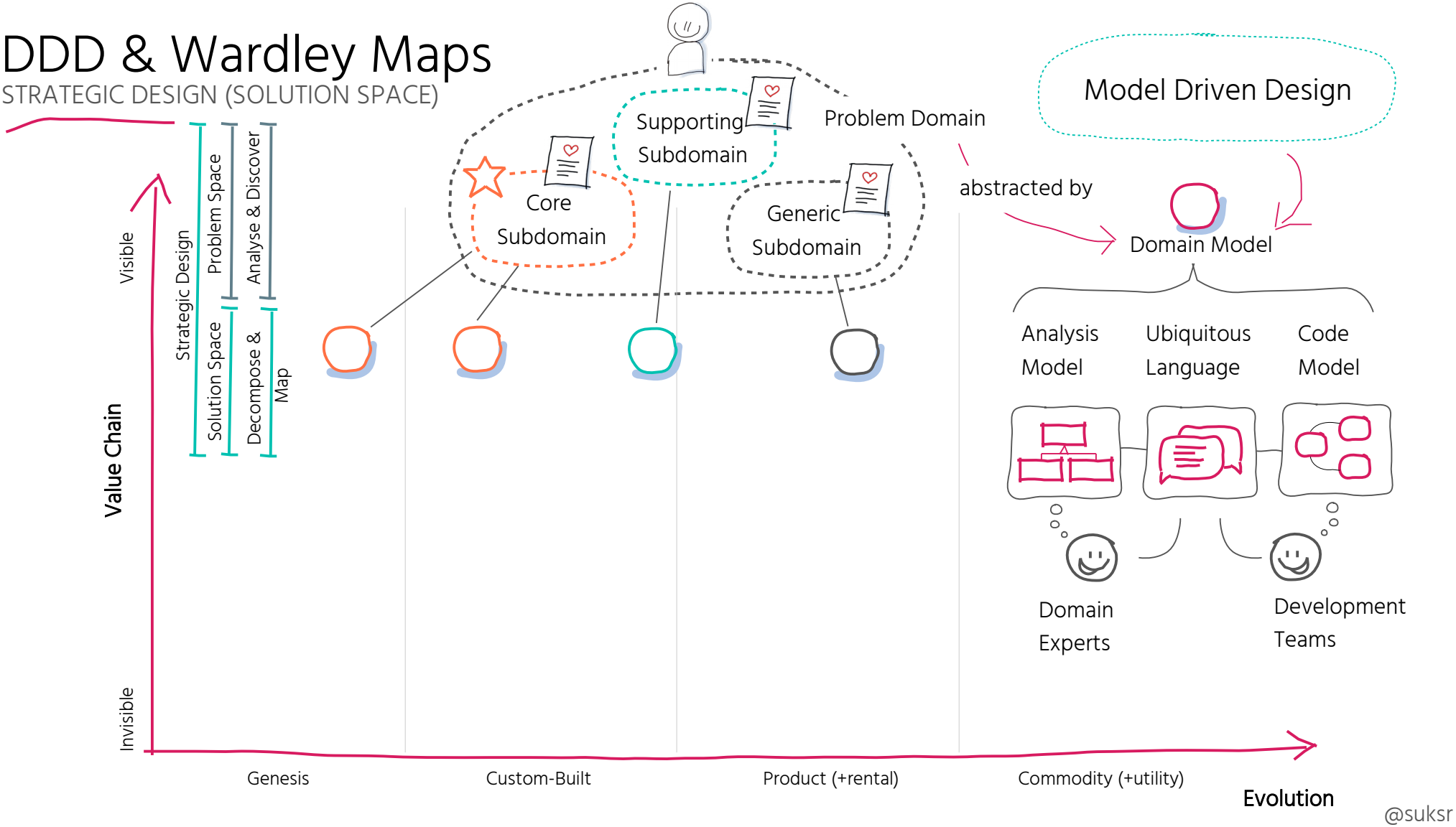
STRATEGIC DESIGN (PROBLEM SPACE)



@suksr

DDD & Wardley Maps

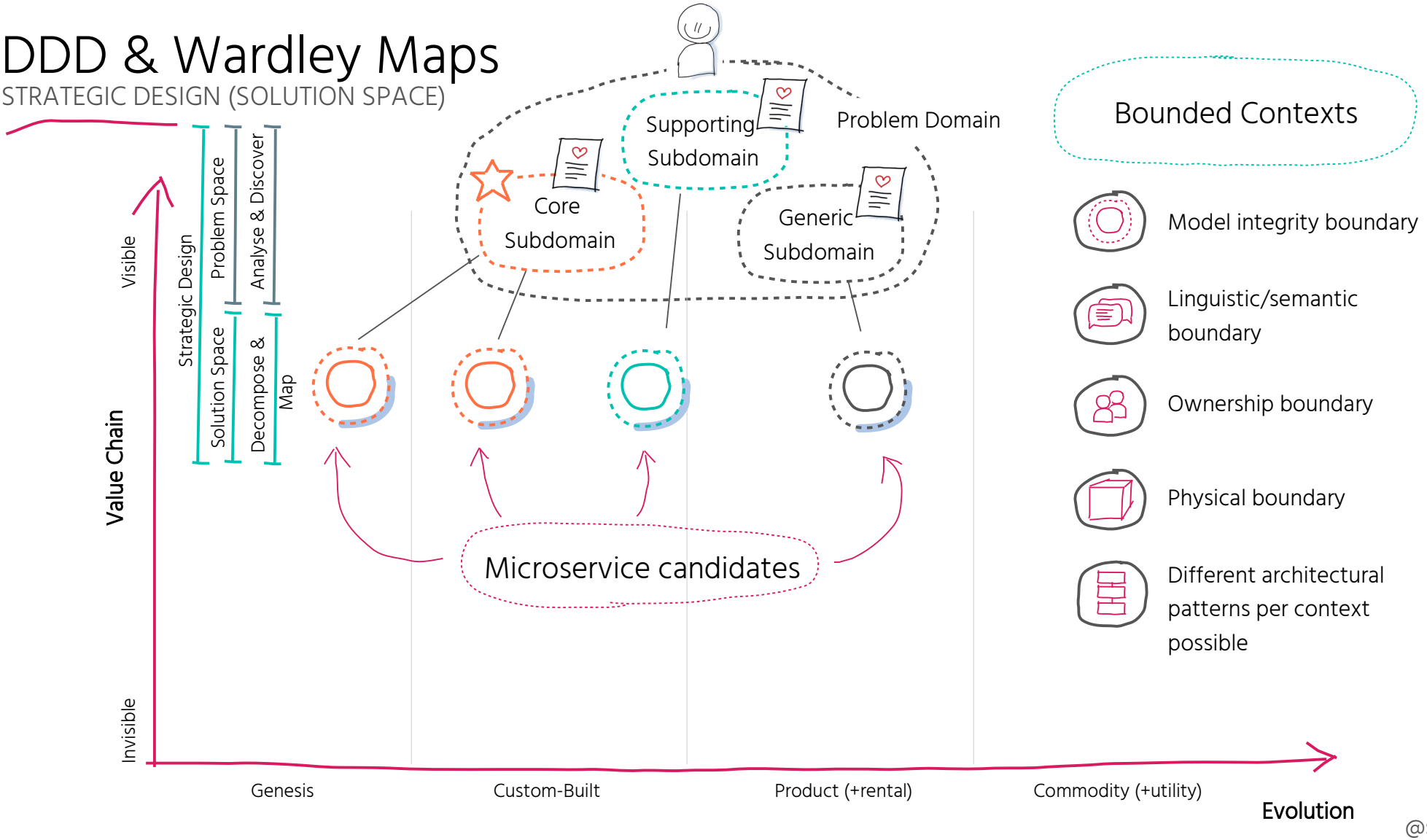
STRATEGIC DESIGN (SOLUTION SPACE)



@suksr

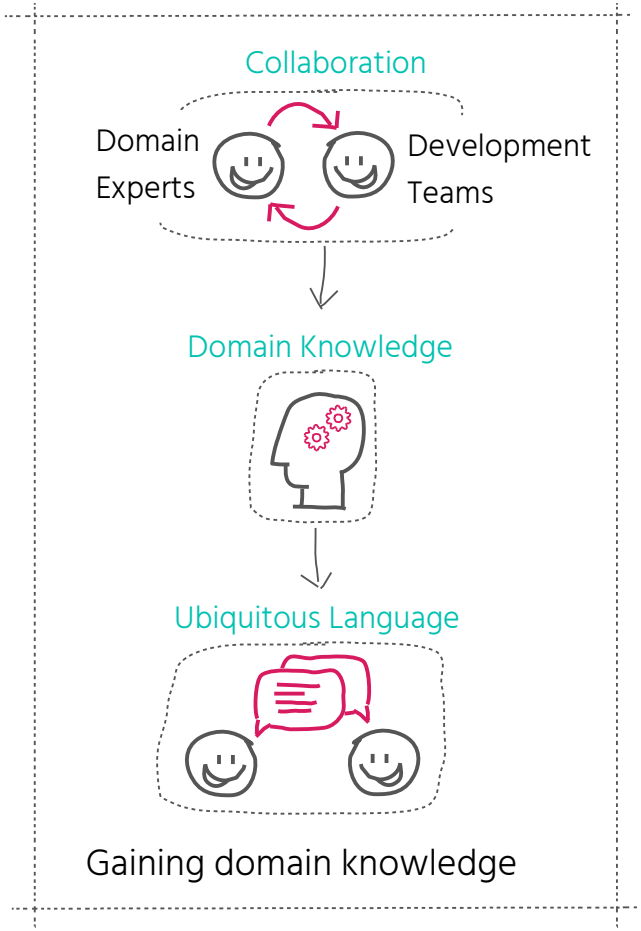
DDD & Wardley Maps

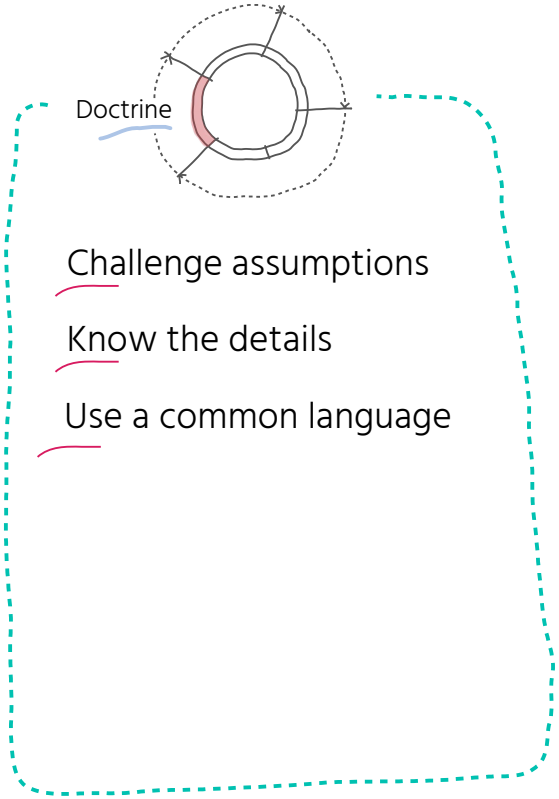
STRATEGIC DESIGN (SOLUTION SPACE)



@suksr

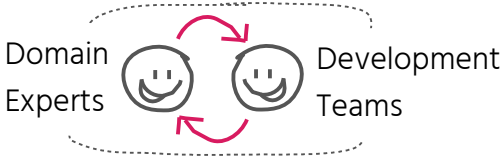
Strategic Design of DDD & Doctrinal Principles





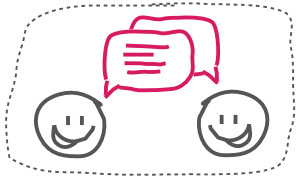
Strategic Design of DDD & Doctrinal Principles

Collaboration

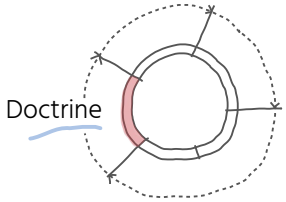


Domain Knowledge

Ubiquitous Language



Gaining domain knowledge

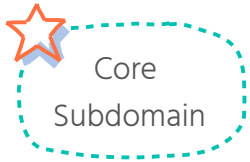


Challenge assumptions

Know the details

Use a common language

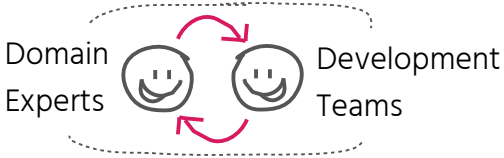
Focus on high situational awareness



Discovering the core subdomain

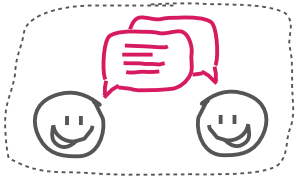
Strategic Design of DDD & Doctrinal Principles

Collaboration

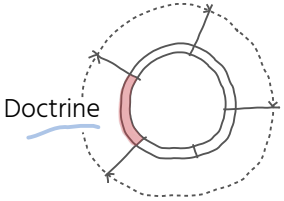


Domain Knowledge

Ubiquitous Language



Gaining domain knowledge



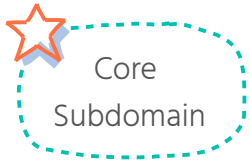
Challenge assumptions

Know the details

Use a common language

Focus on high situational awareness

Think small (as in contracts)



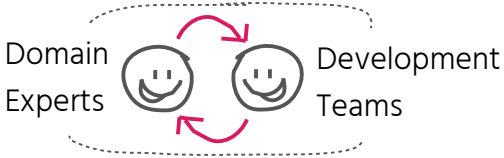
Discovering the core subdomain

Decomposing into modular components (Bounded Contexts)

@suksr

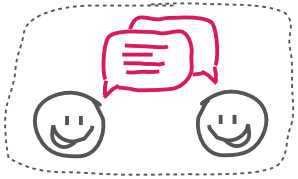
Strategic Design of DDD & Doctrinal Principles

Collaboration

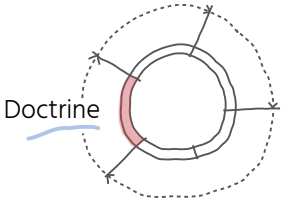


Domain Knowledge

Ubiquitous Language



Gaining domain knowledge



Challenge assumptions

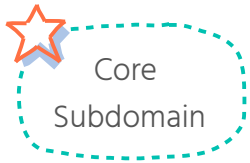
Know the details

Use a common language

Focus on high situational awareness

Think small (as in contracts)

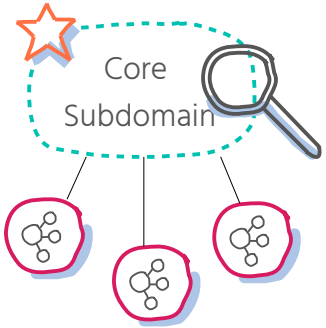
Use appropriate methods



Discovering the core subdomain

Decomposing into modular components (Bounded Contexts)

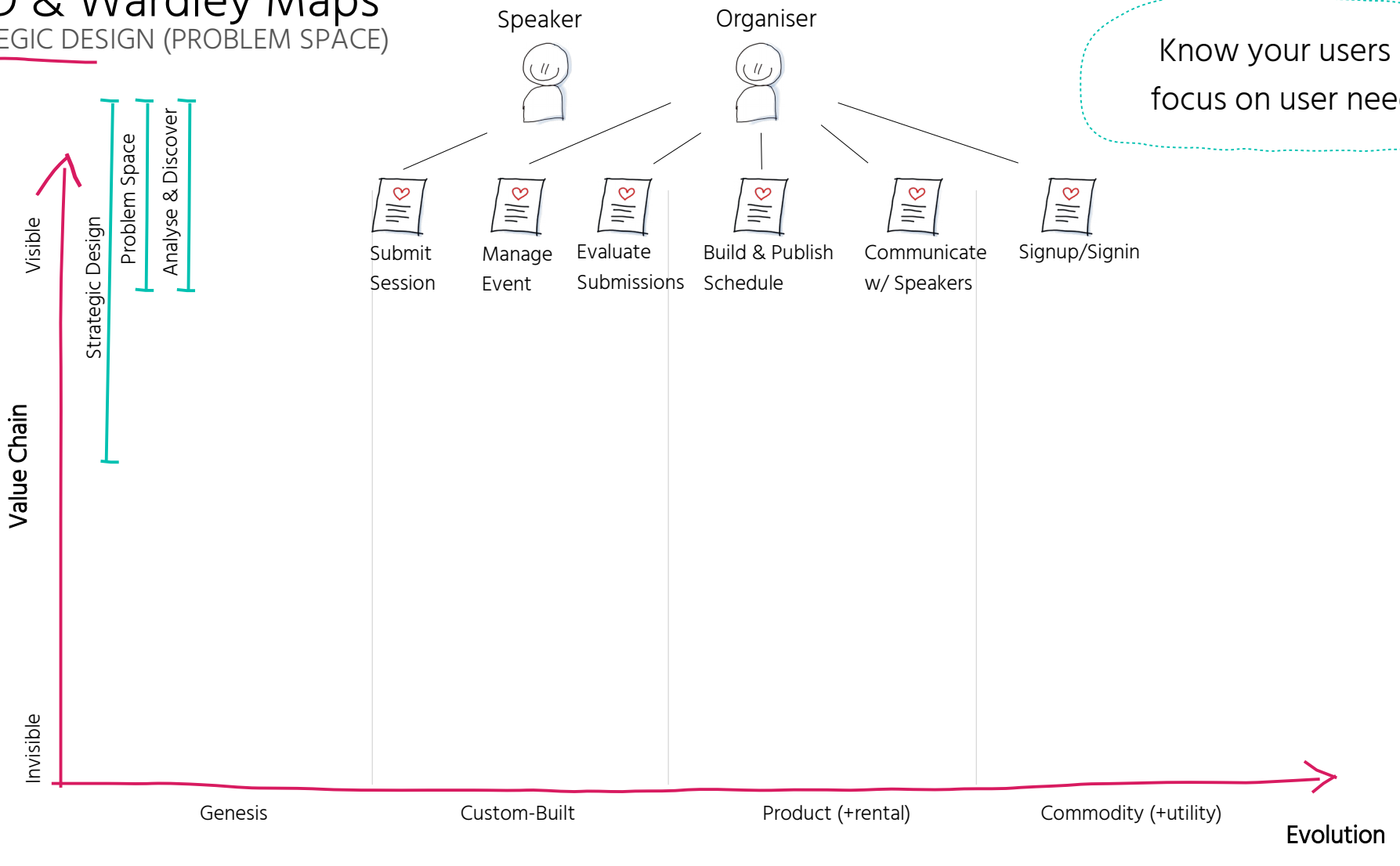
DDD suits best for the core subdomain



@suksr

DDD & Wardley Maps

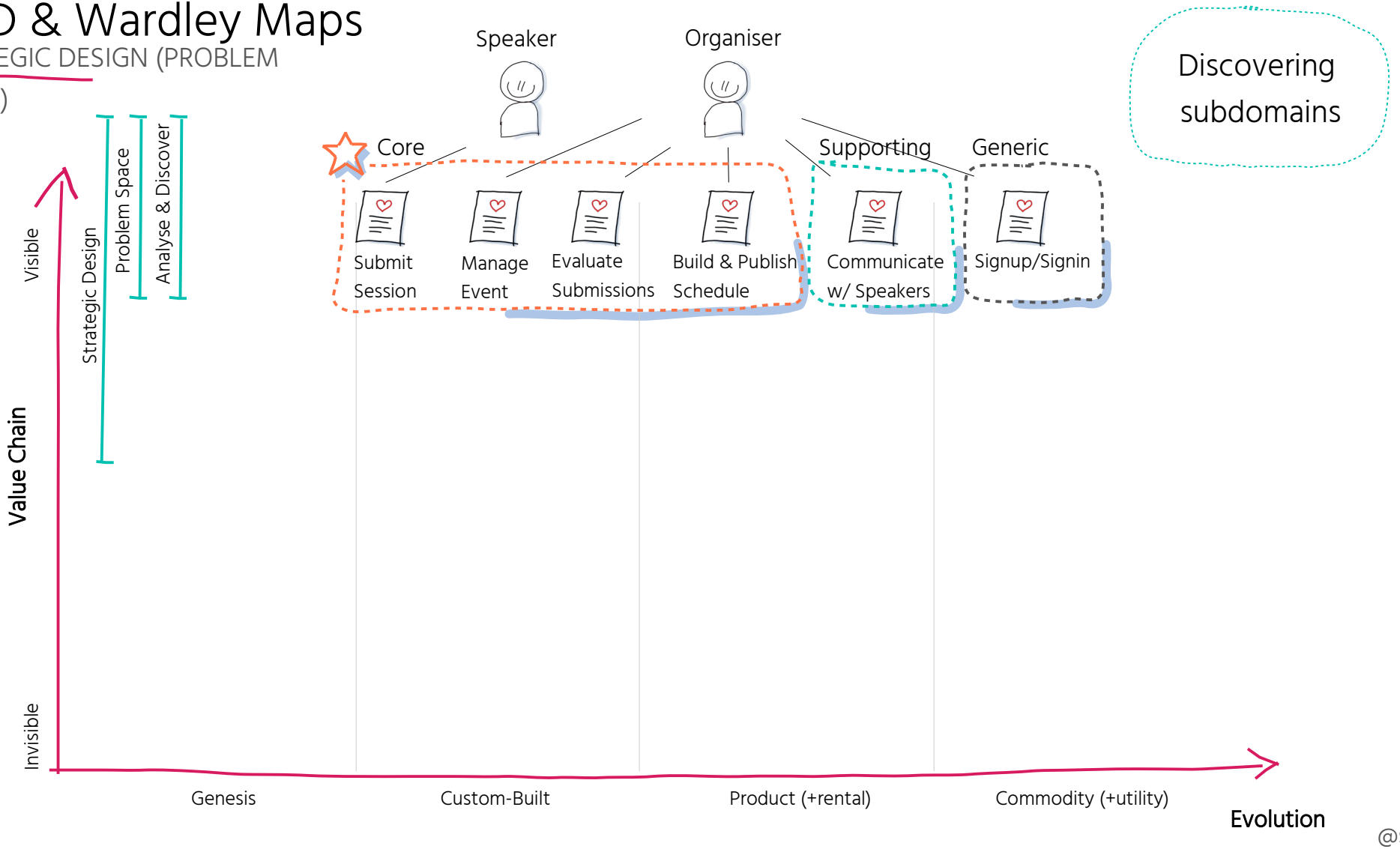
STRATEGIC DESIGN (PROBLEM SPACE)



@suksr

DDD & Wardley Maps

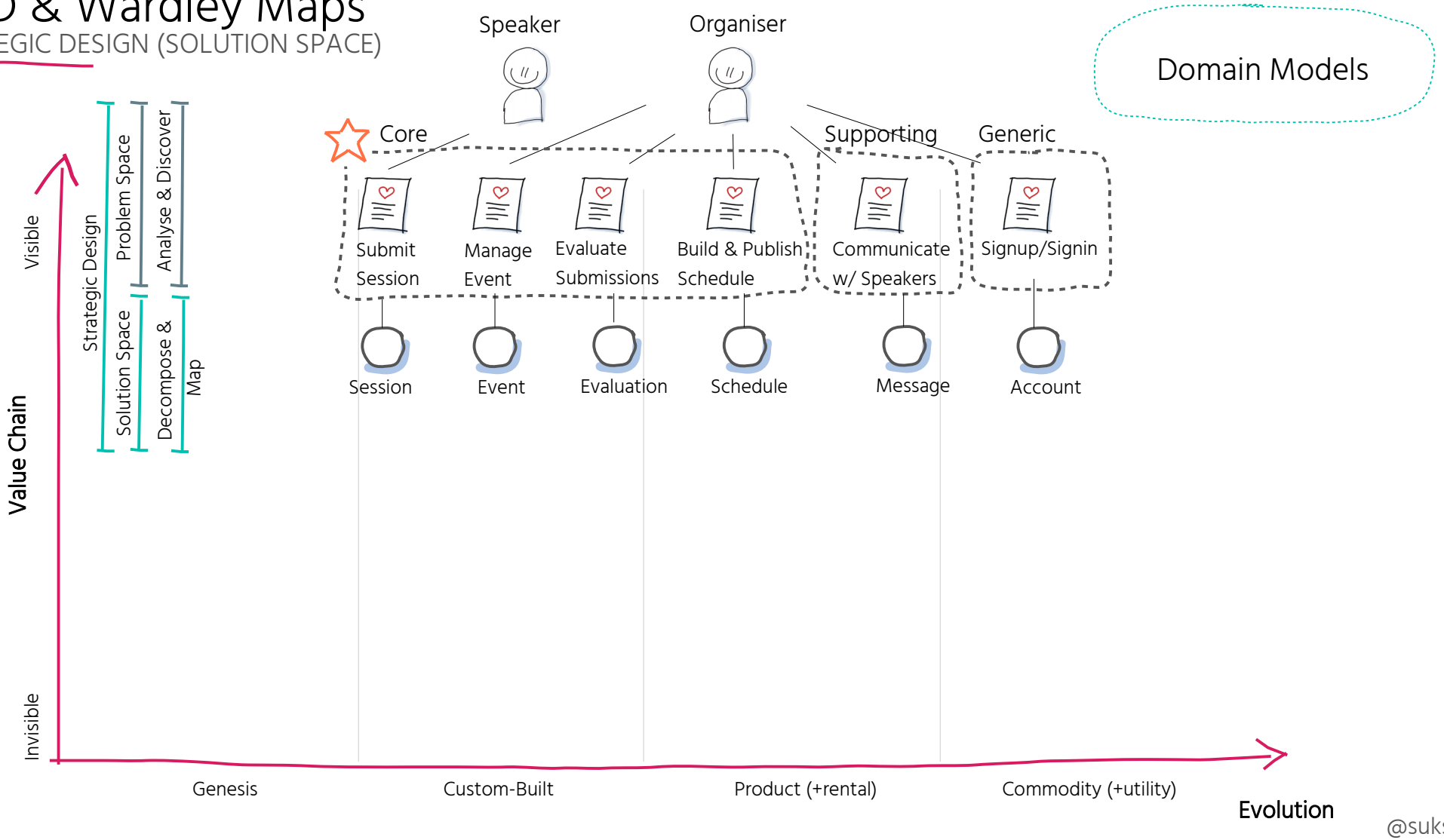
STRATEGIC DESIGN (PROBLEM SPACE)



@suksr

DDD & Wardley Maps

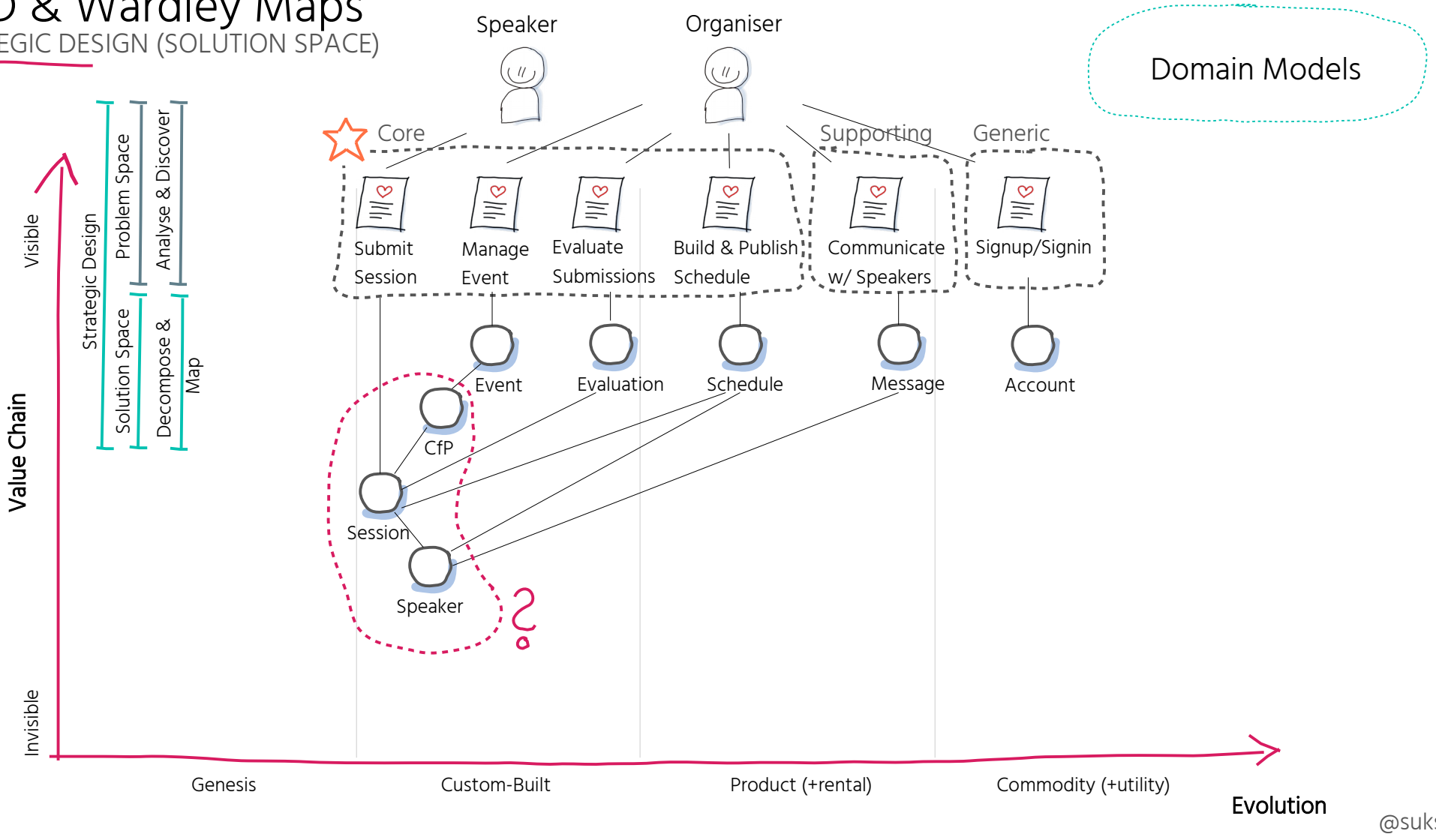
STRATEGIC DESIGN (SOLUTION SPACE)



@suksr

DDD & Wardley Maps

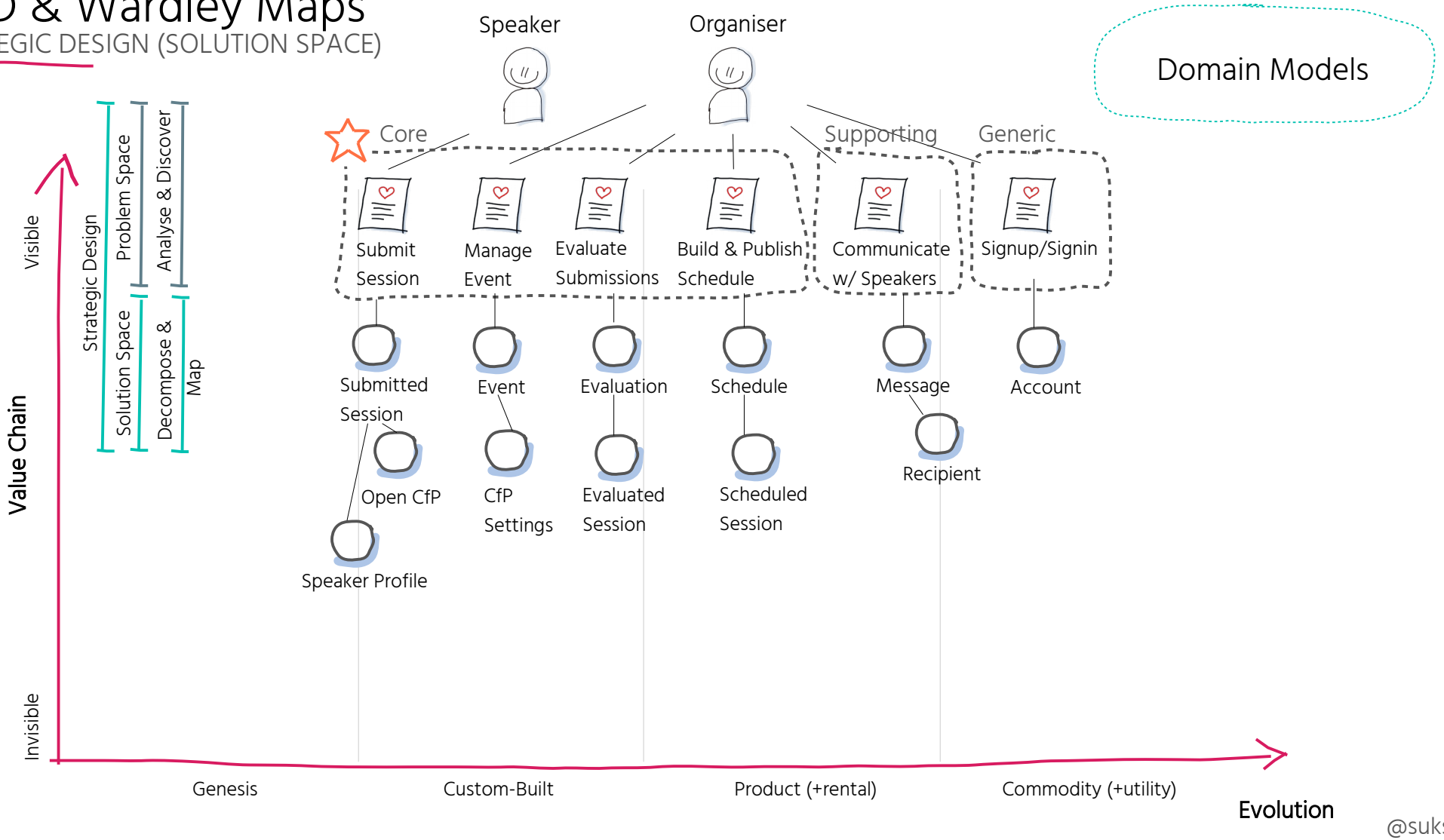
STRATEGIC DESIGN (SOLUTION SPACE)



@suksr

DDD & Wardley Maps

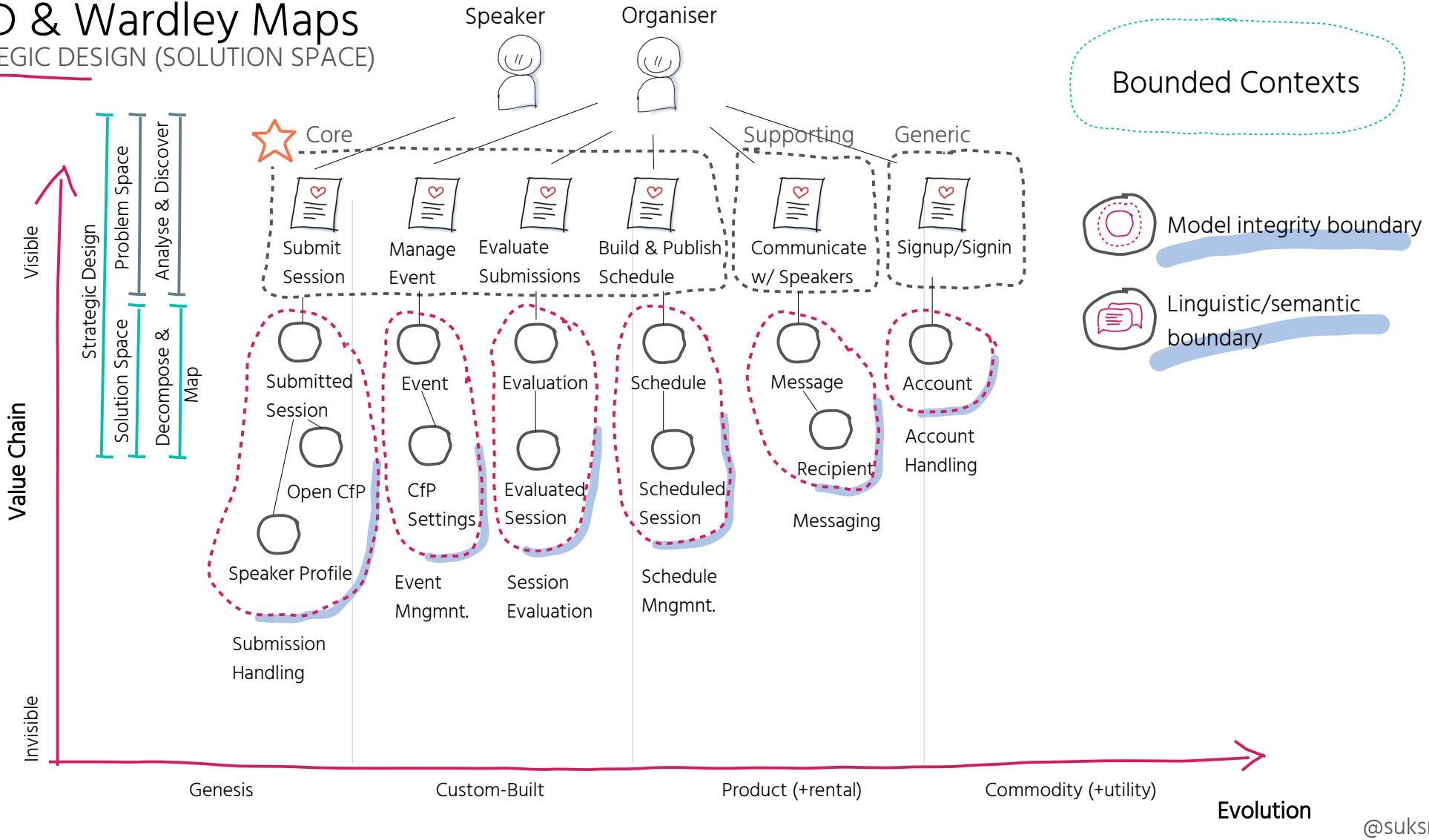
STRATEGIC DESIGN (SOLUTION SPACE)



@suksr

DDD & Wardley Maps

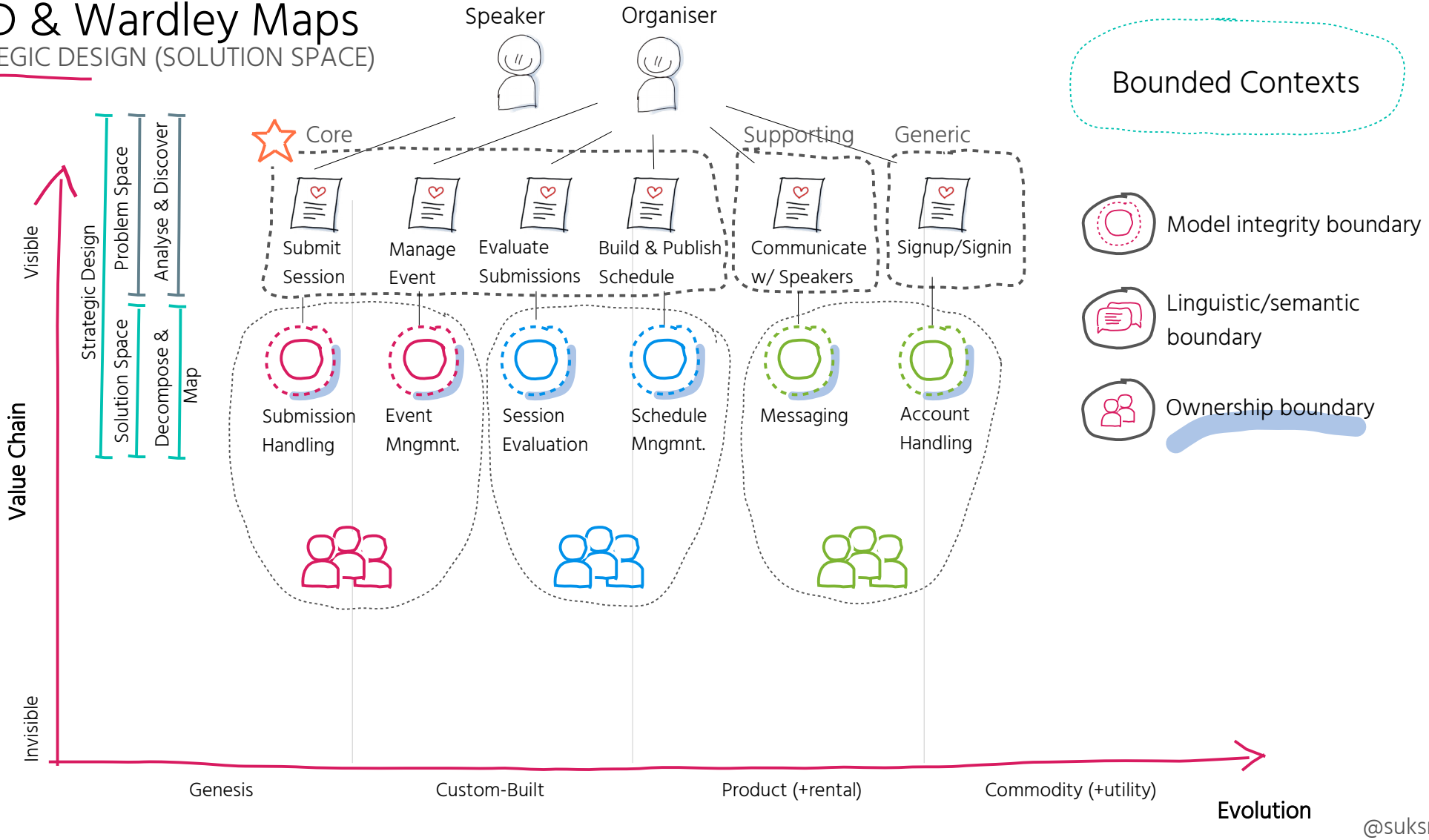
STRATEGIC DESIGN (SOLUTION SPACE)



@suksr

DDD & Wardley Maps

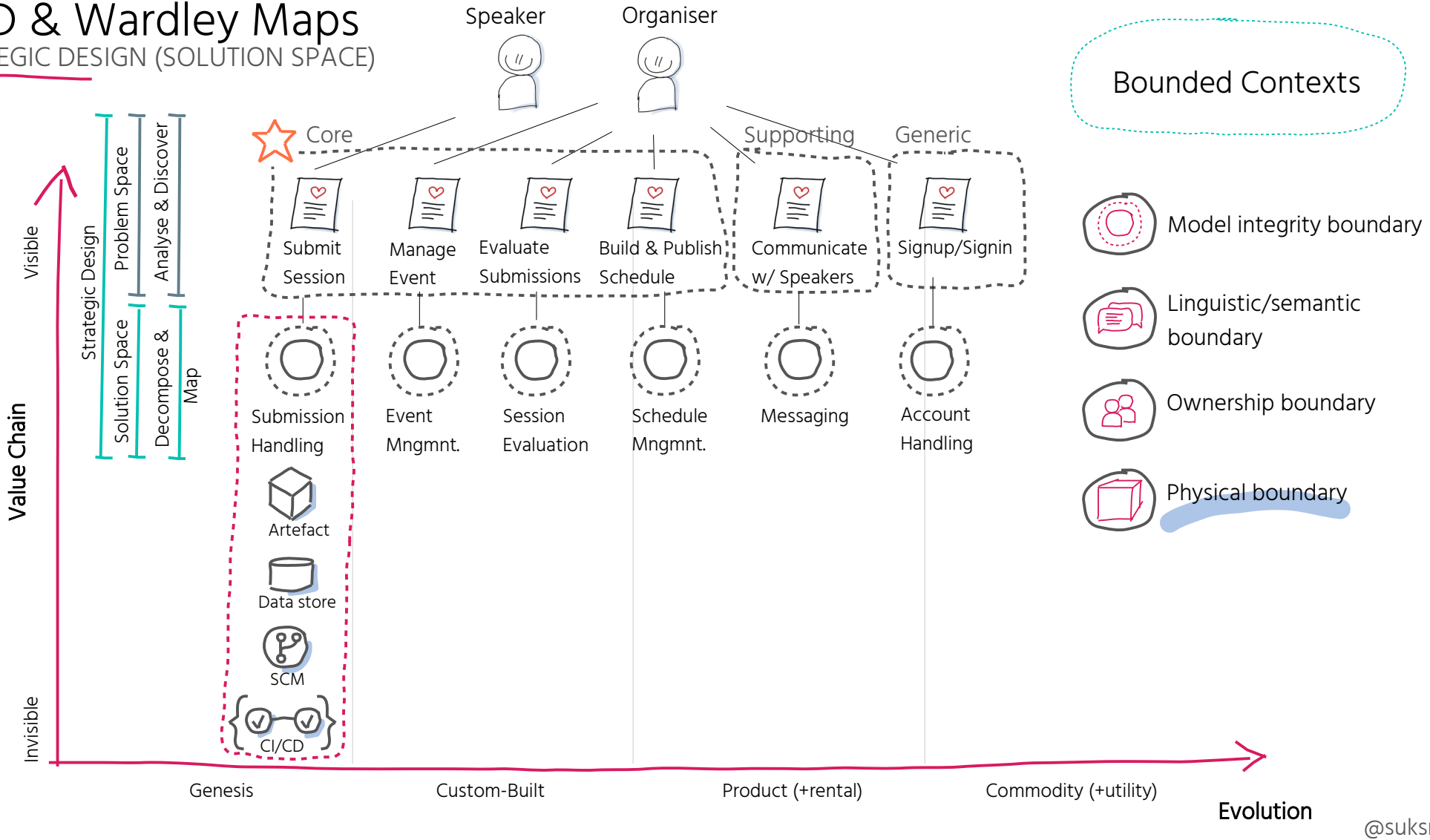
STRATEGIC DESIGN (SOLUTION SPACE)



@suksr

DDD & Wardley Maps

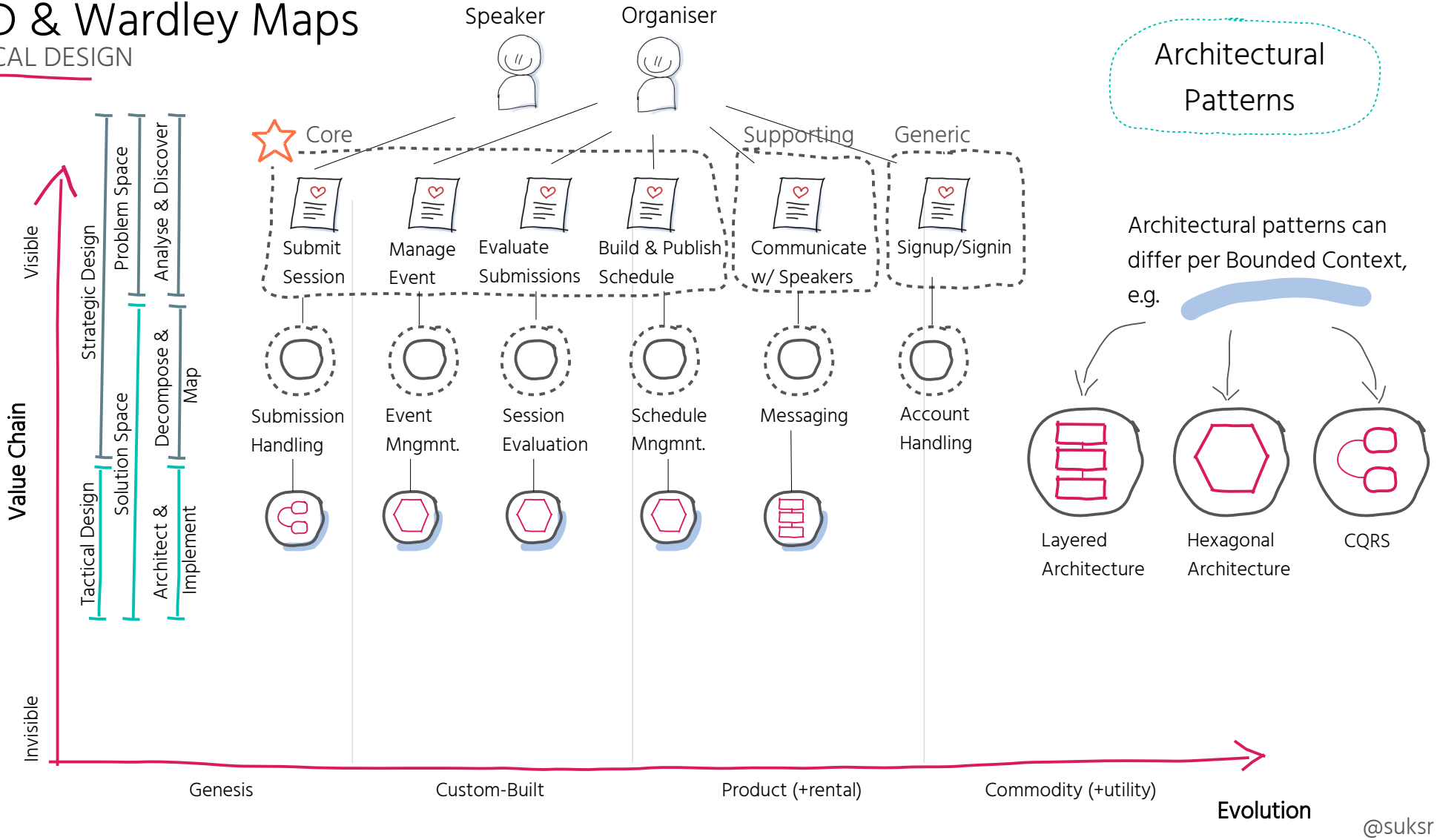
STRATEGIC DESIGN (SOLUTION SPACE)



@suksr

DDD & Wardley Maps

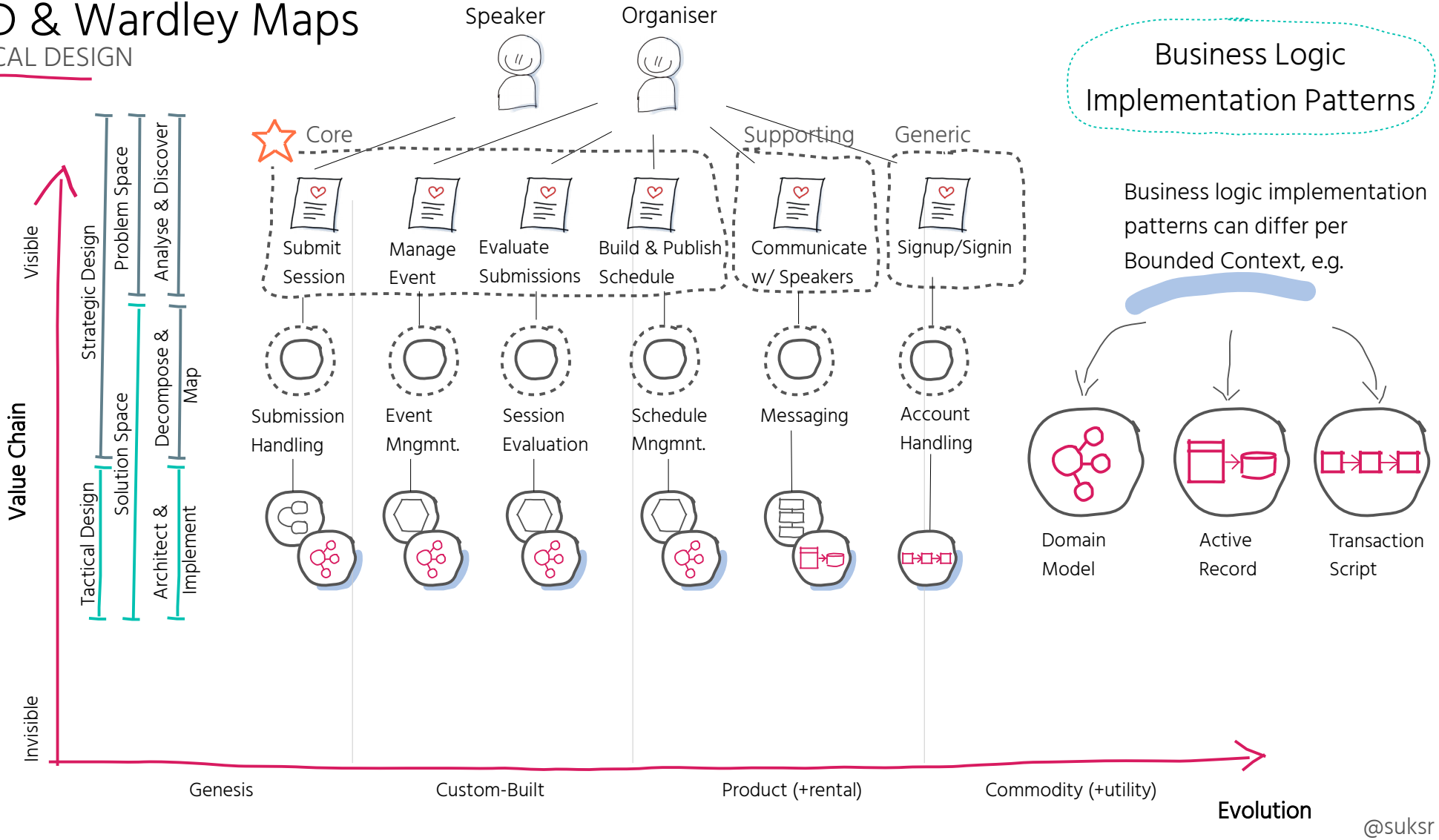
TACTICAL DESIGN



@suksr

DDD & Wardley Maps

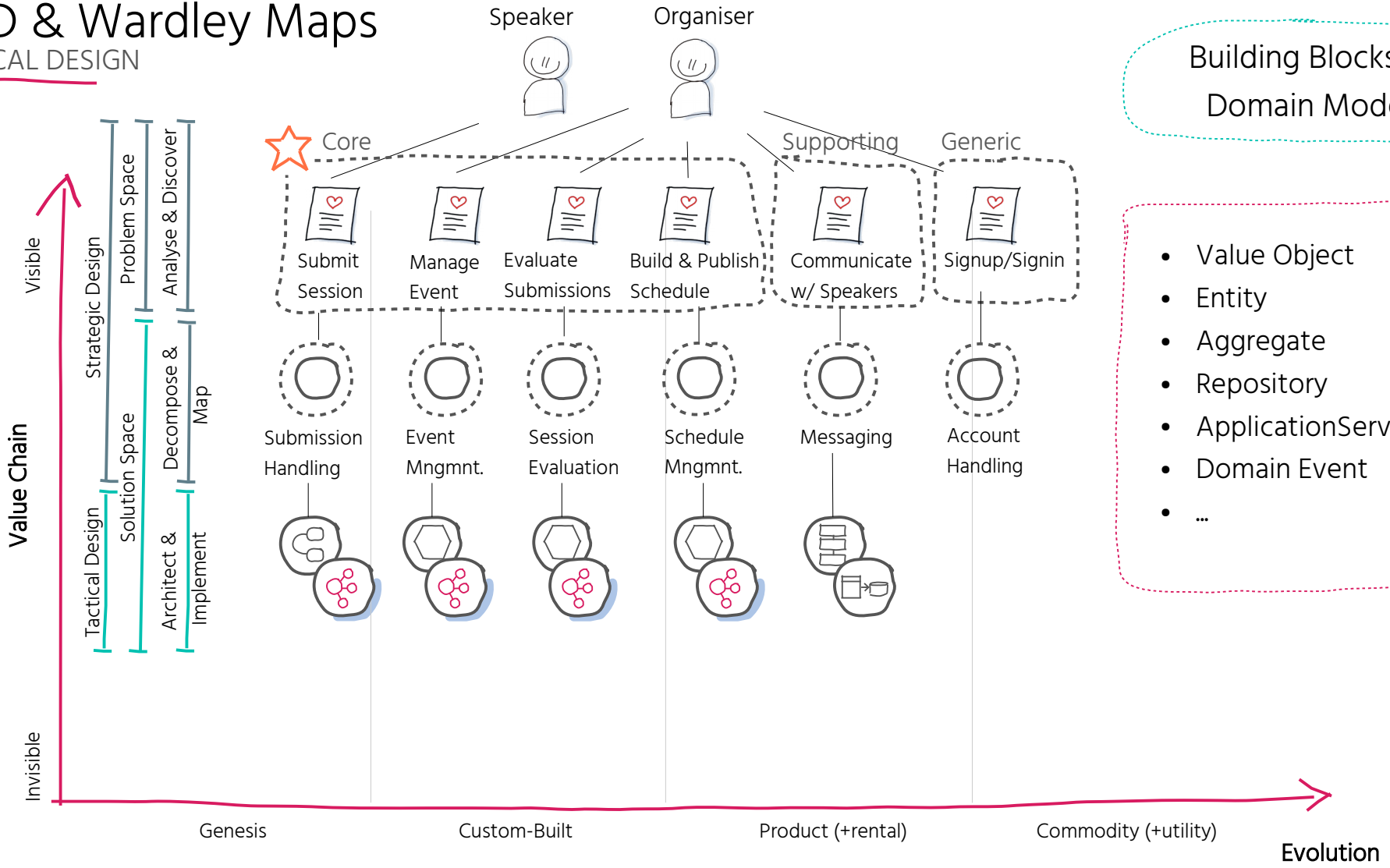
TACTICAL DESIGN



@suksr

DDD & Wardley Maps

TACTICAL DESIGN



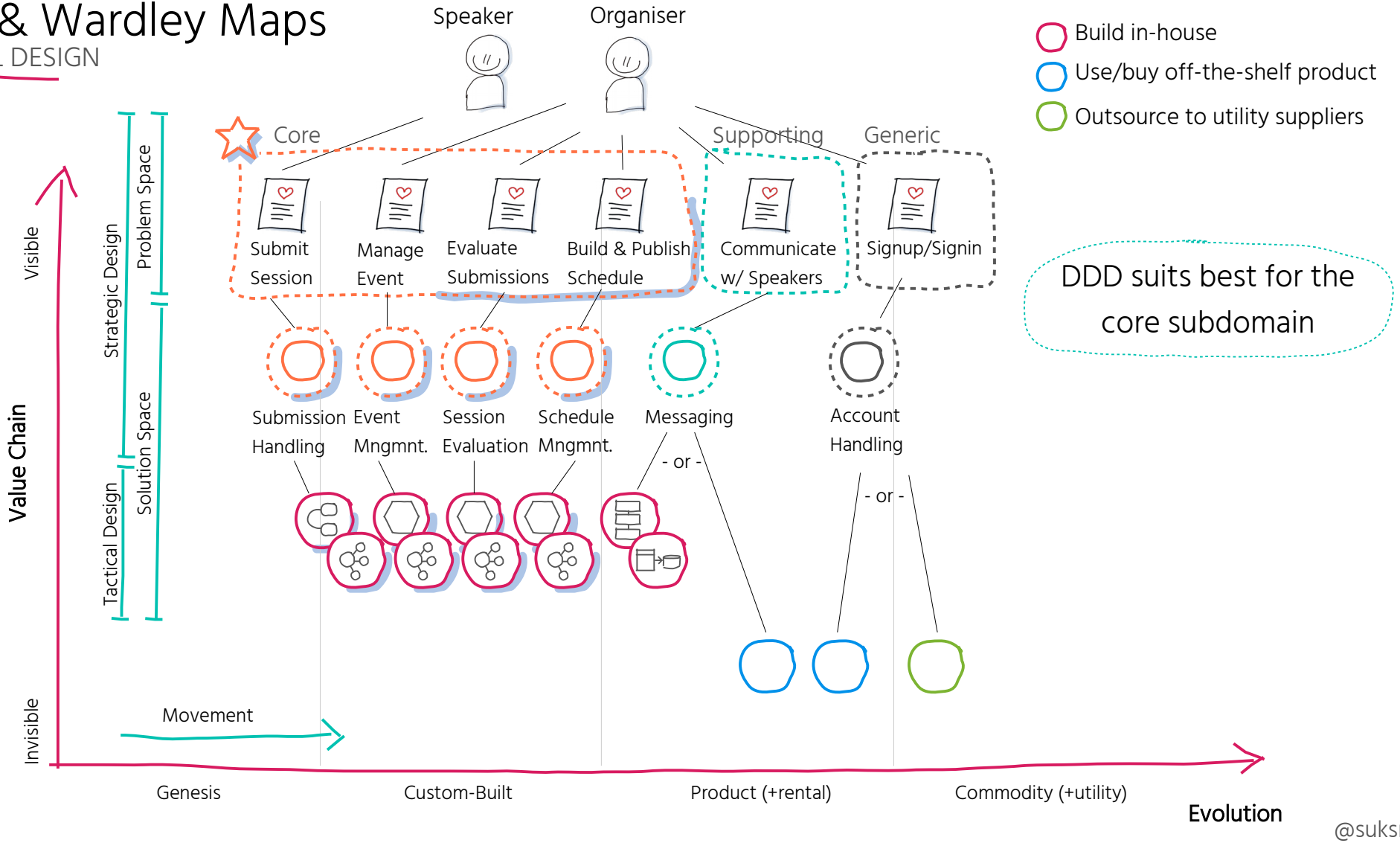
Building Blocks of Domain Models

- Value Object
- Entity
- Aggregate
- Repository
- ApplicationService
- Domain Event
- ...

@suksr

DDD & Wardley Maps

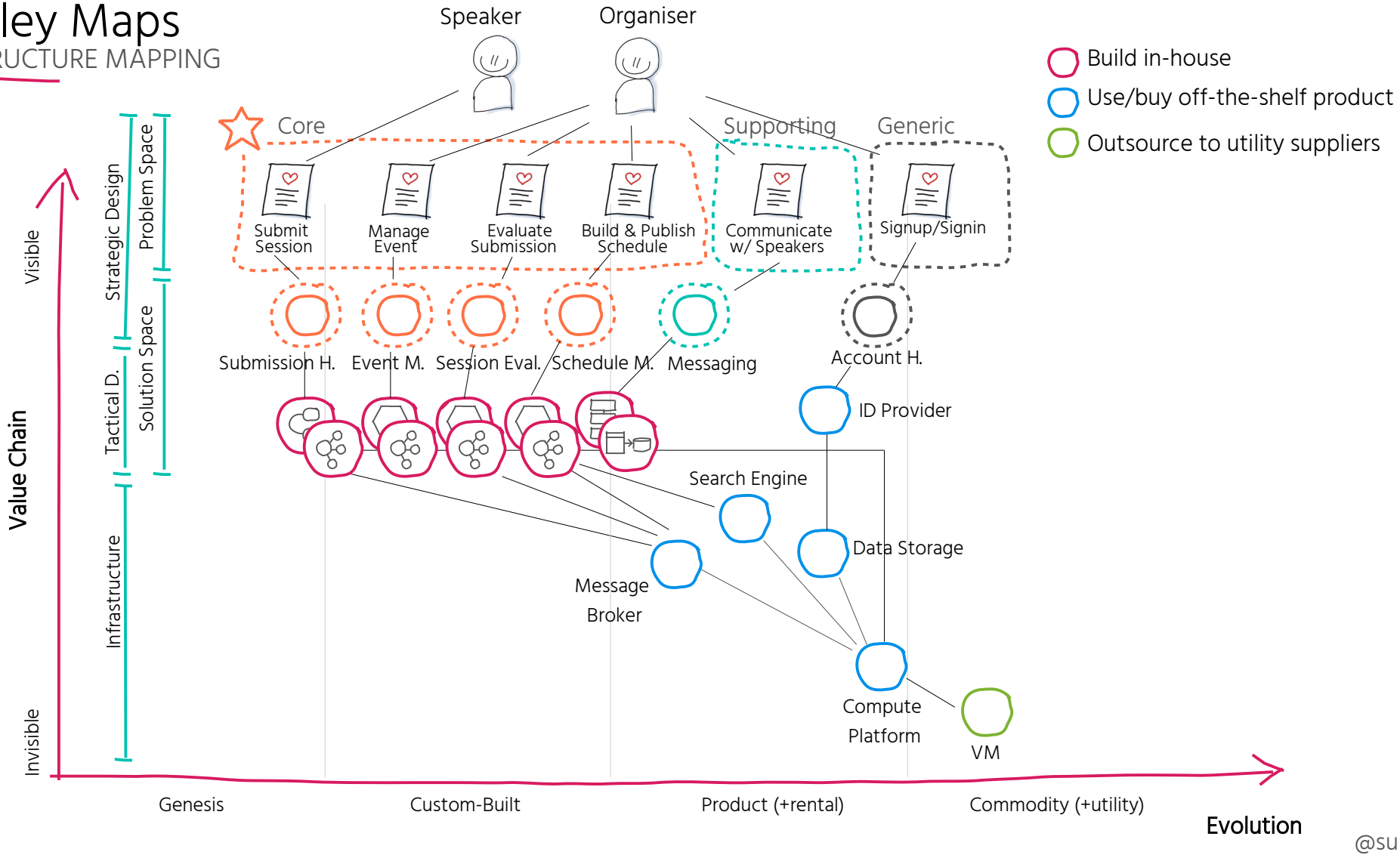
TACTICAL DESIGN



@suksr

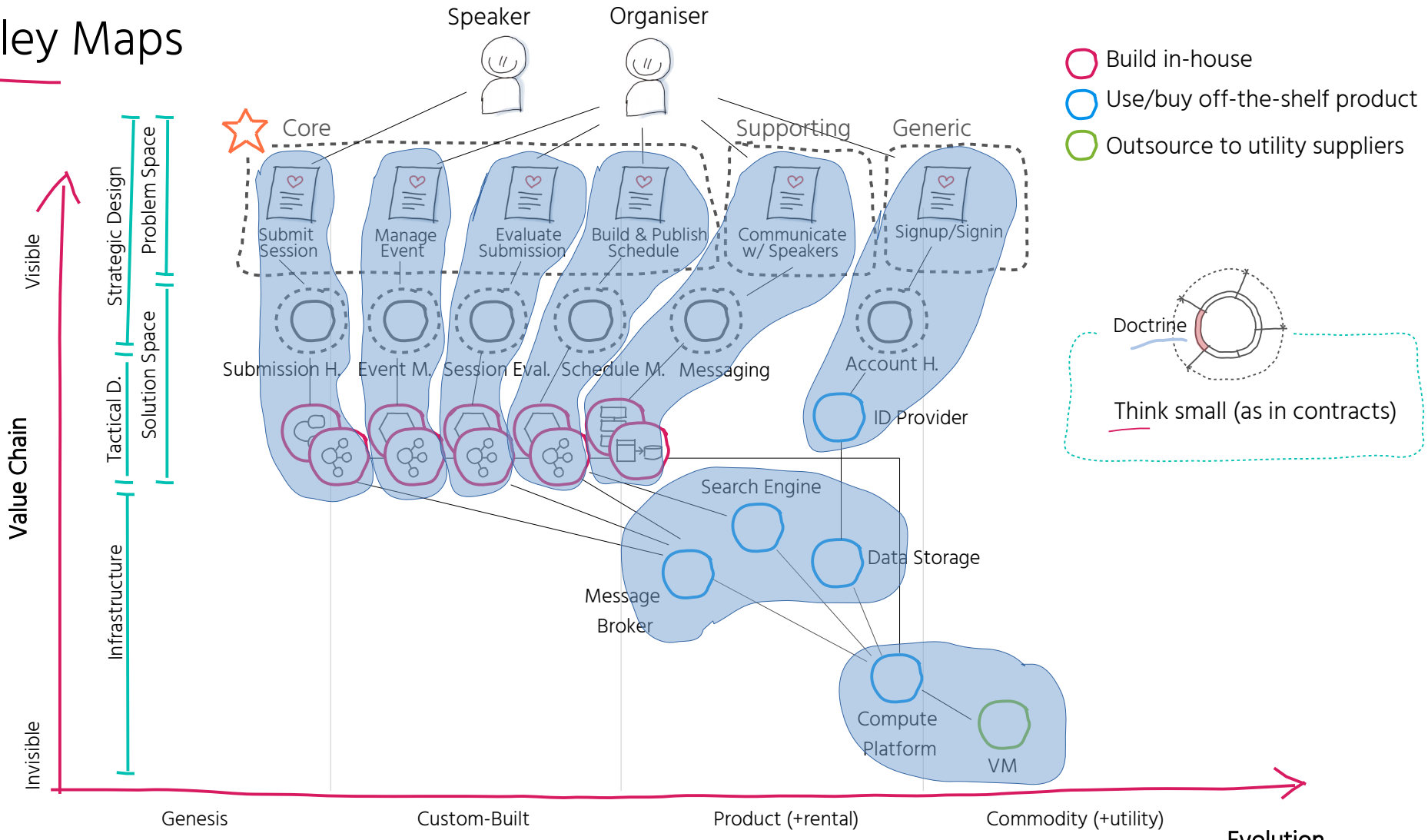
Wardley Maps

INFRASTRUCTURE MAPPING



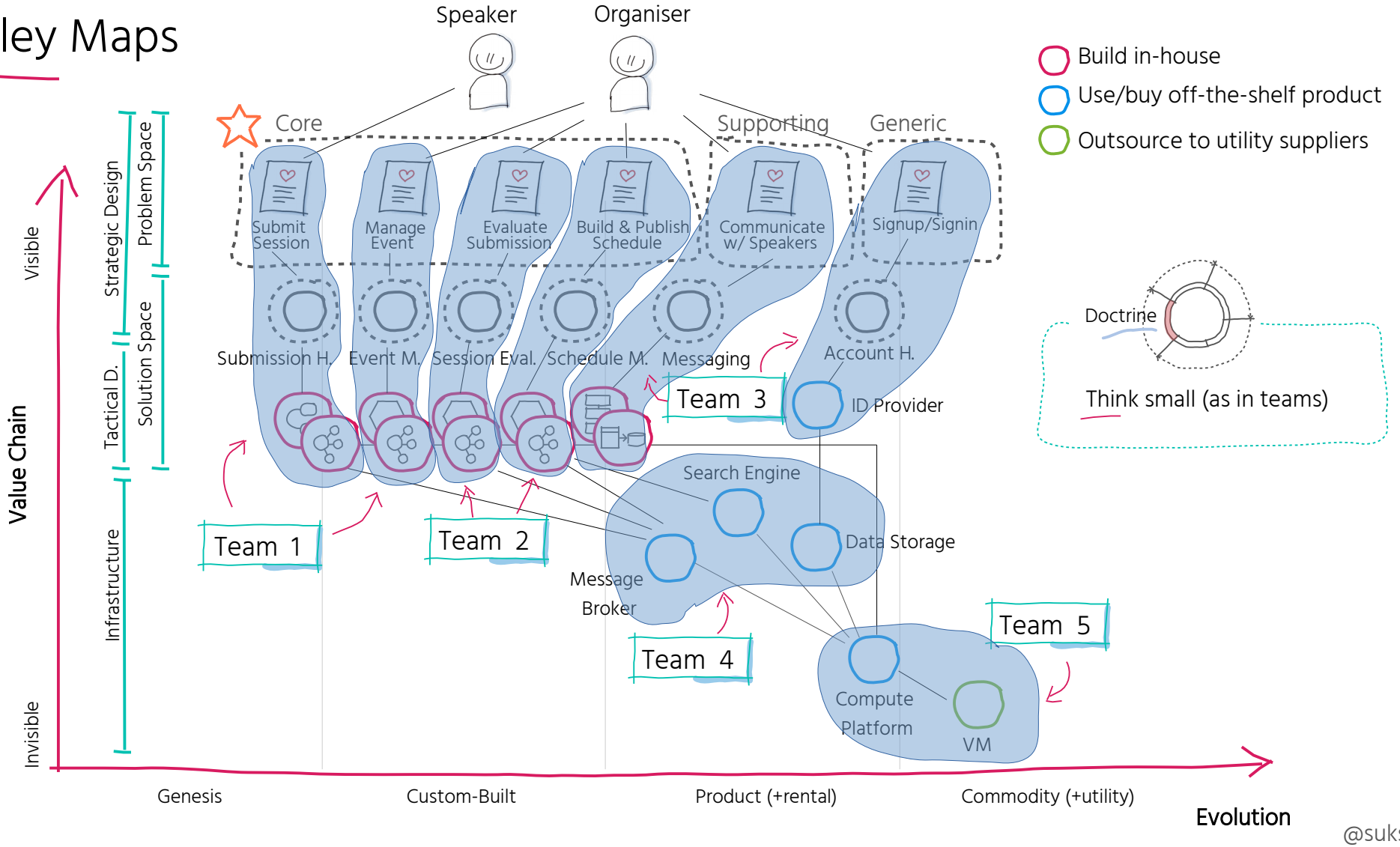
@suksr

Wardley Maps



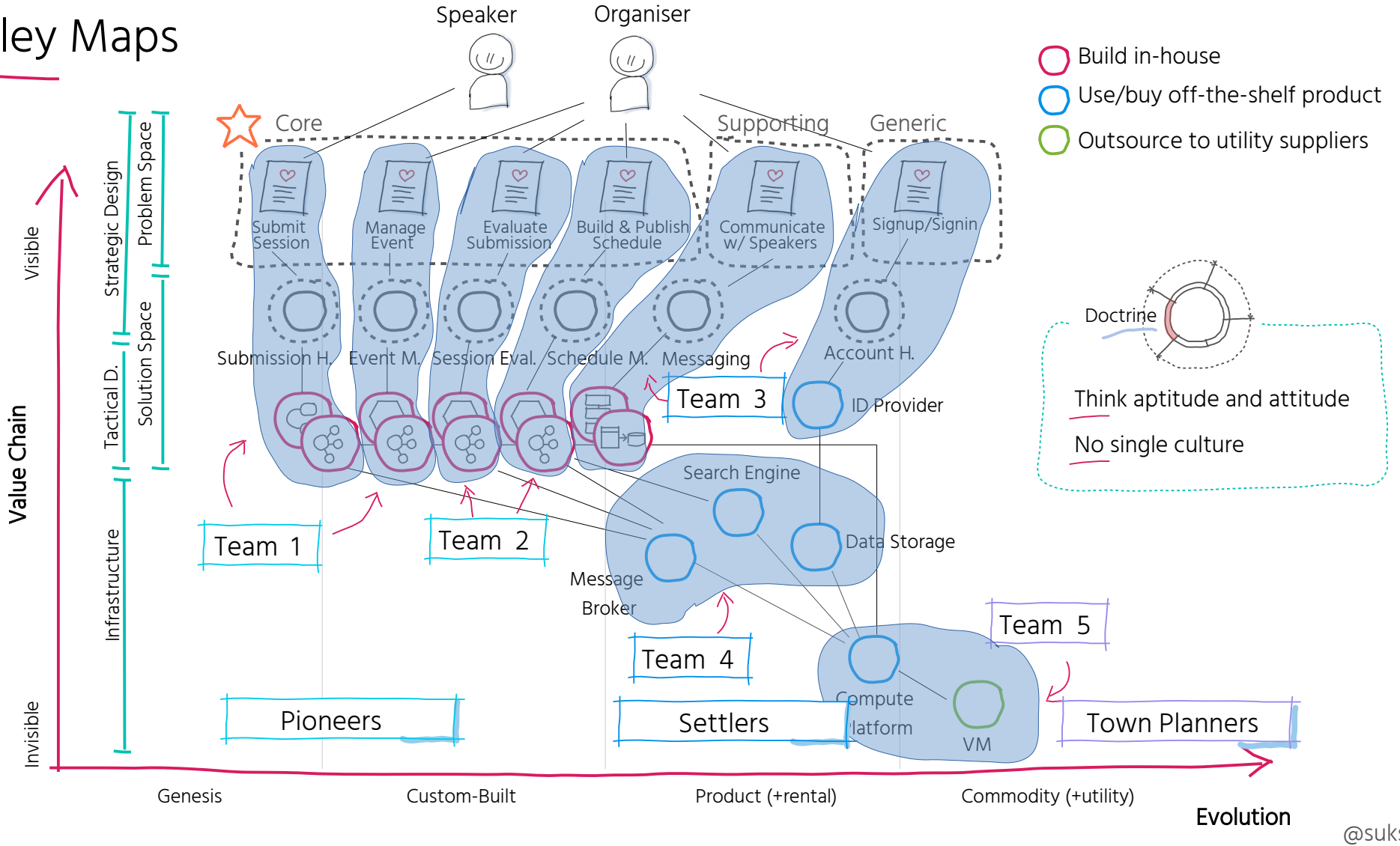
@suksr

Wardley Maps



@suksr

Wardley Maps

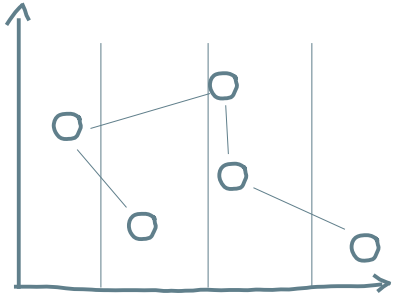


@suksr

3 Perspectives to Build Adaptive Systems

1.

Business-Strategy



w/ Wardley Maps

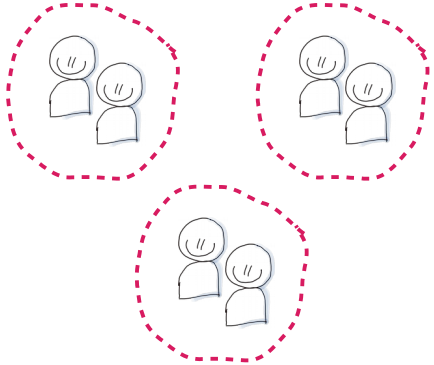
2.

Software-Design/
-Architecture

w/ Domain-Driven Design

3.

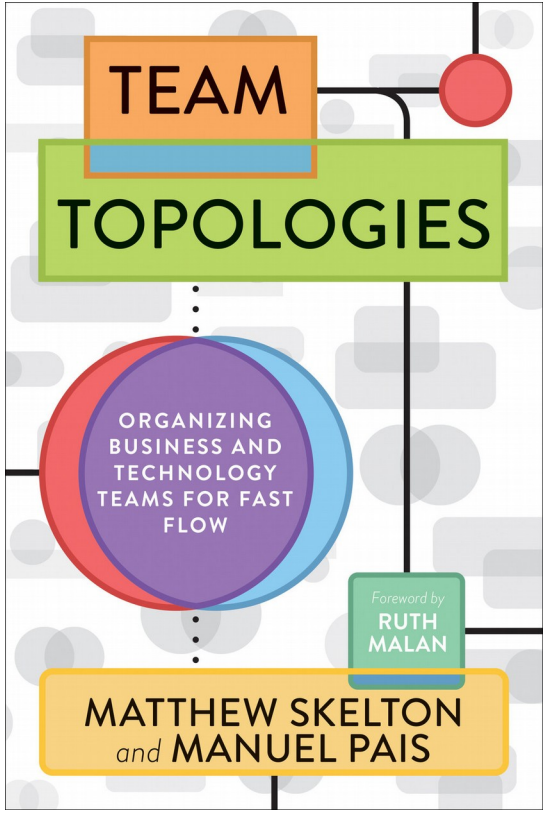
Team-Organization



w/ Team Topologies

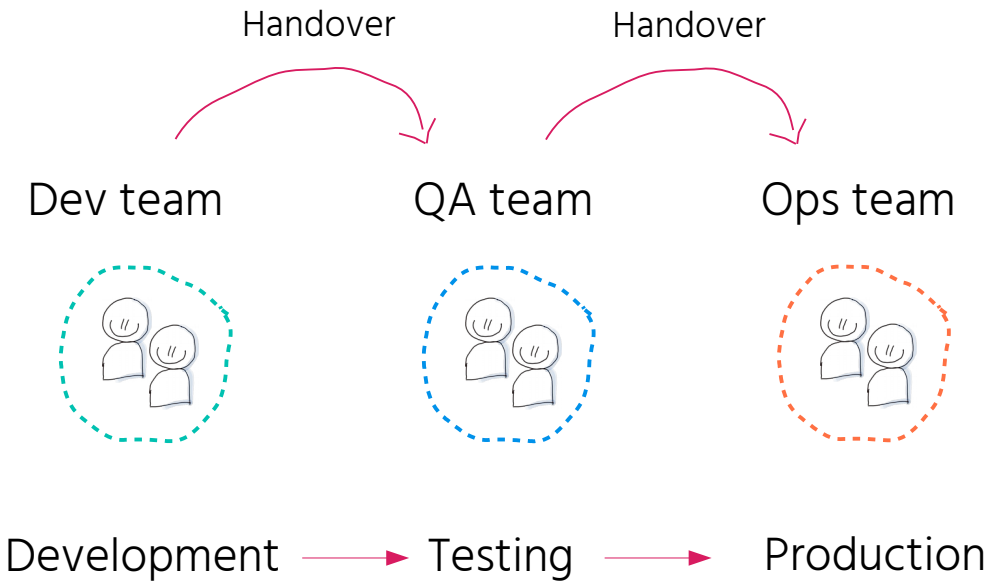
@suksr

Team Topologies



"Overall, the Team Topologies approach advocates for organization design that optimizes for flow of change and feedback from running systems."

Linear sequence w/ functional silos not optimized for flow

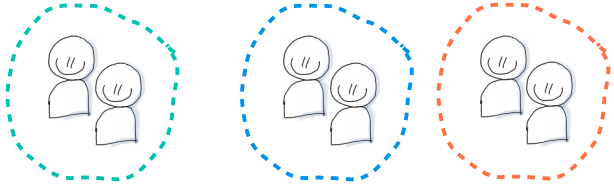


To optimize for flow of change requires …

cross-functional, autonomous teams

no handover between teams

restricting communication between teams

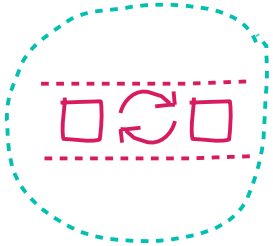


small, long-lived teams

minimizing cognitive load

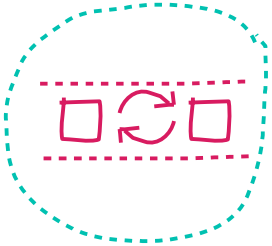
team ownership

Four Team Types



Stream-aligned team

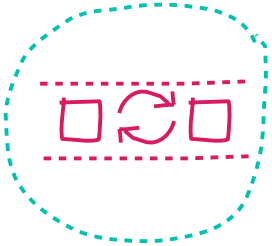
Four Team Types



Stream-aligned team

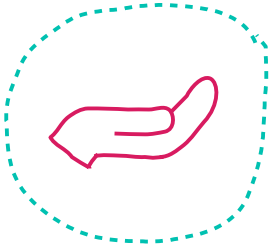
Platform team

Four Team Types



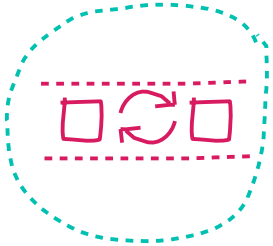
Stream-aligned team

Platform team



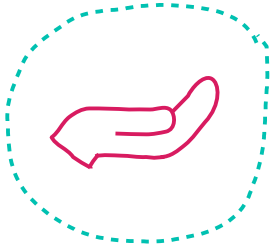
Enabling team

Four Team Types

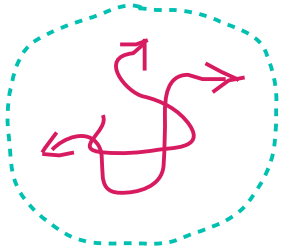


Stream-aligned team

Platform team

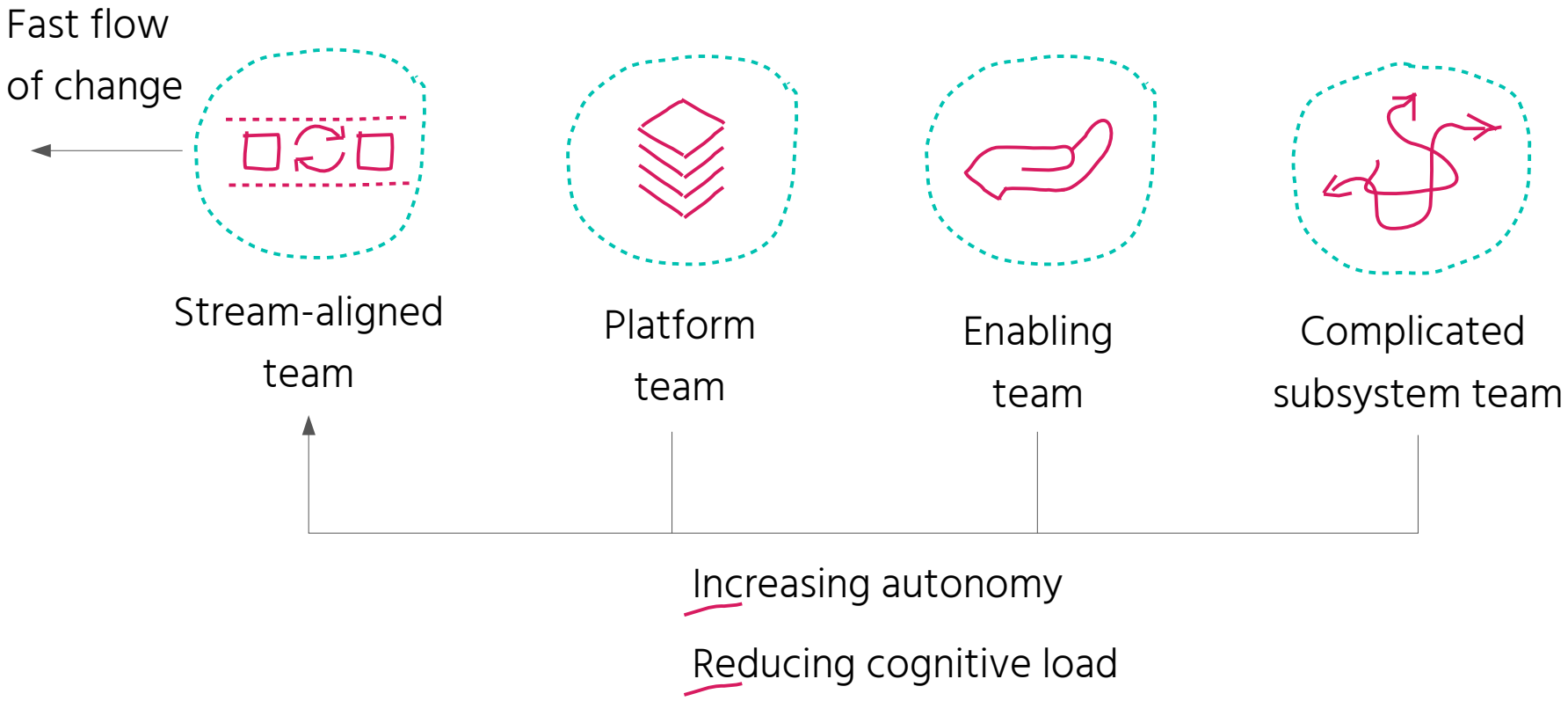


Enabling team



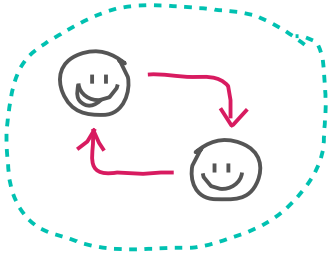
Complicated subsystem team

Four Team Types



@suksr

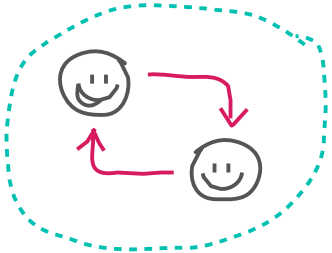
Three Interaction Types



Collaboration

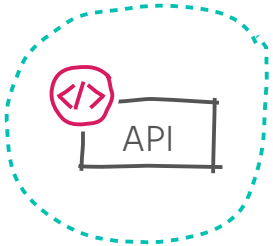
Rapid discovery

Three Interaction Types



Collaboration

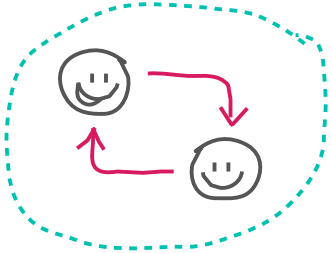
Rapid discovery



X-as-a-Service

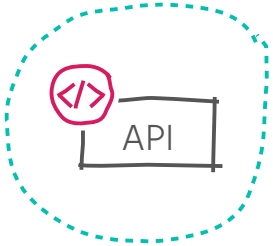
Predictable delivery

Three Interaction Types



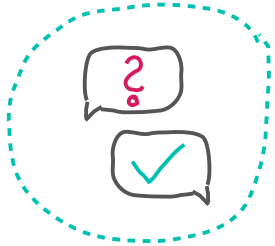
Collaboration

Rapid discovery



X-as-a-Service

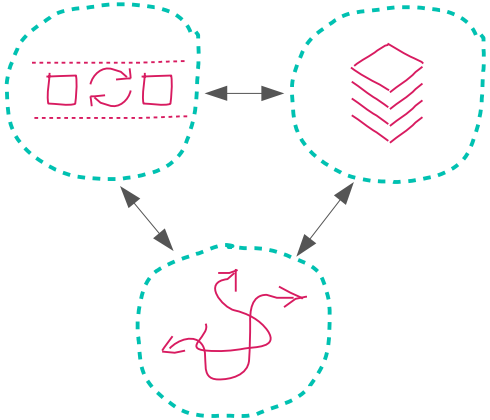
Predictable delivery



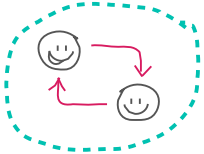
Facilitating

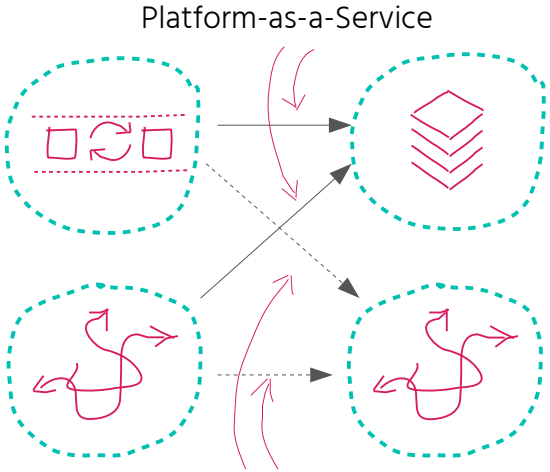
Active help

Intermittent Interactions between Team Types

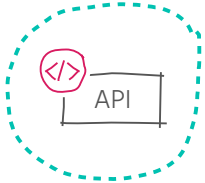


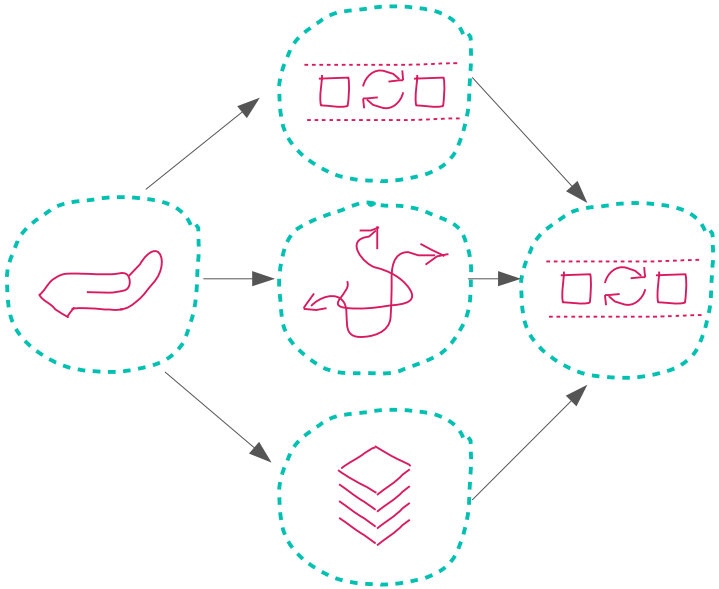
Collaboration





X-as-a-Service



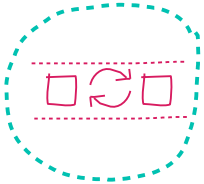


Facilitating

@suksr

Team Topologies & Doctrine

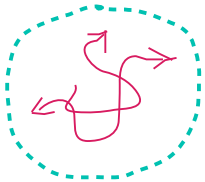
Stream-aligned team



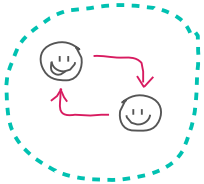
Platform team

Enabling team

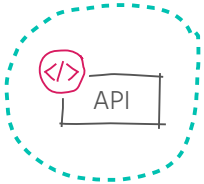
Complicated subsystem team



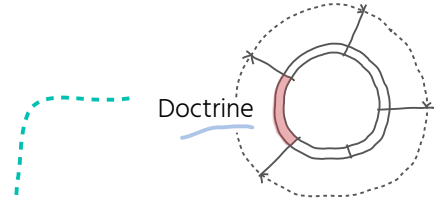
Collaboration



X-as-a-Service



Facilitating



Think small teams

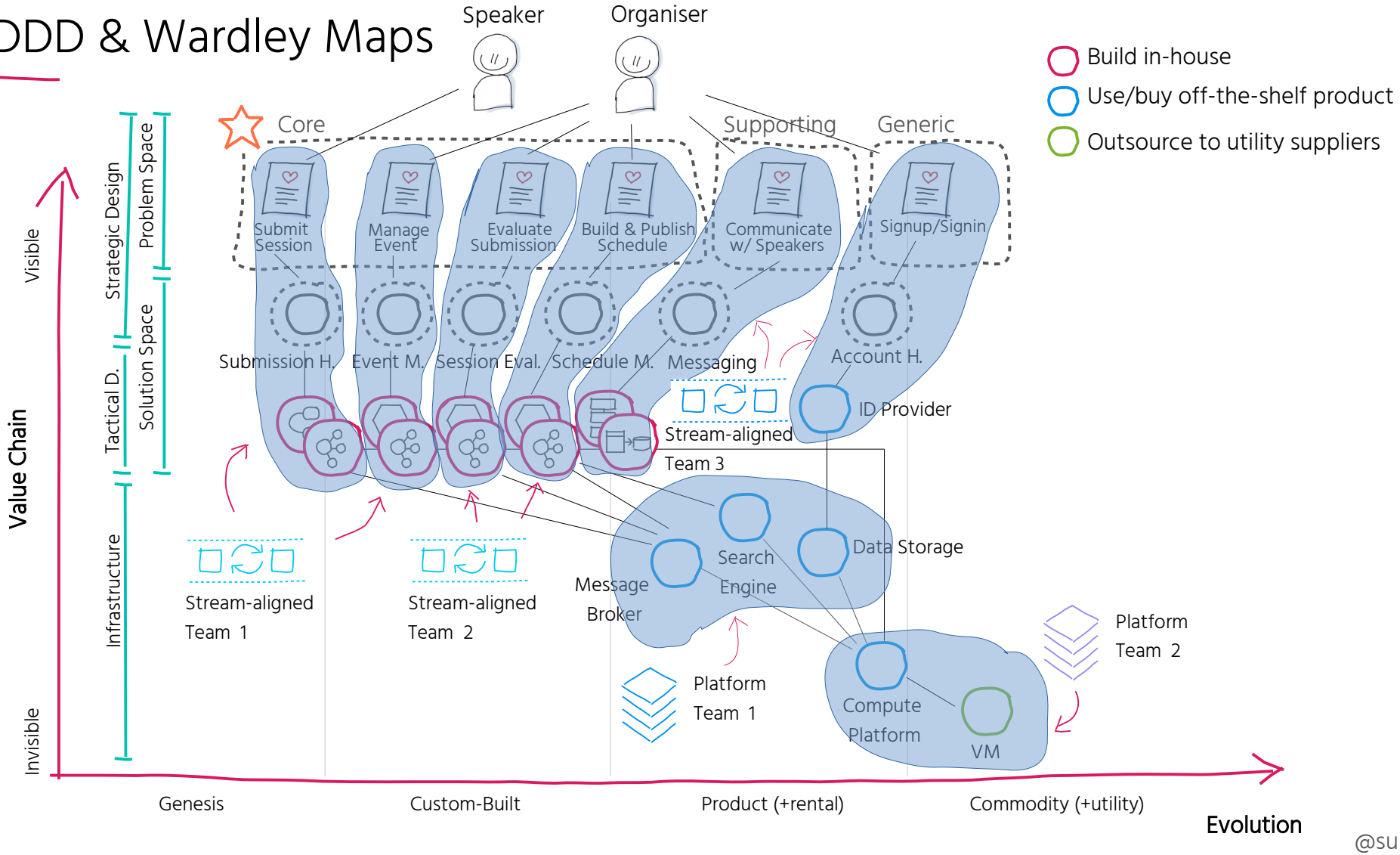
Optimize flow

Provide purpose,
mastery & autonomy

Design for constant evolution

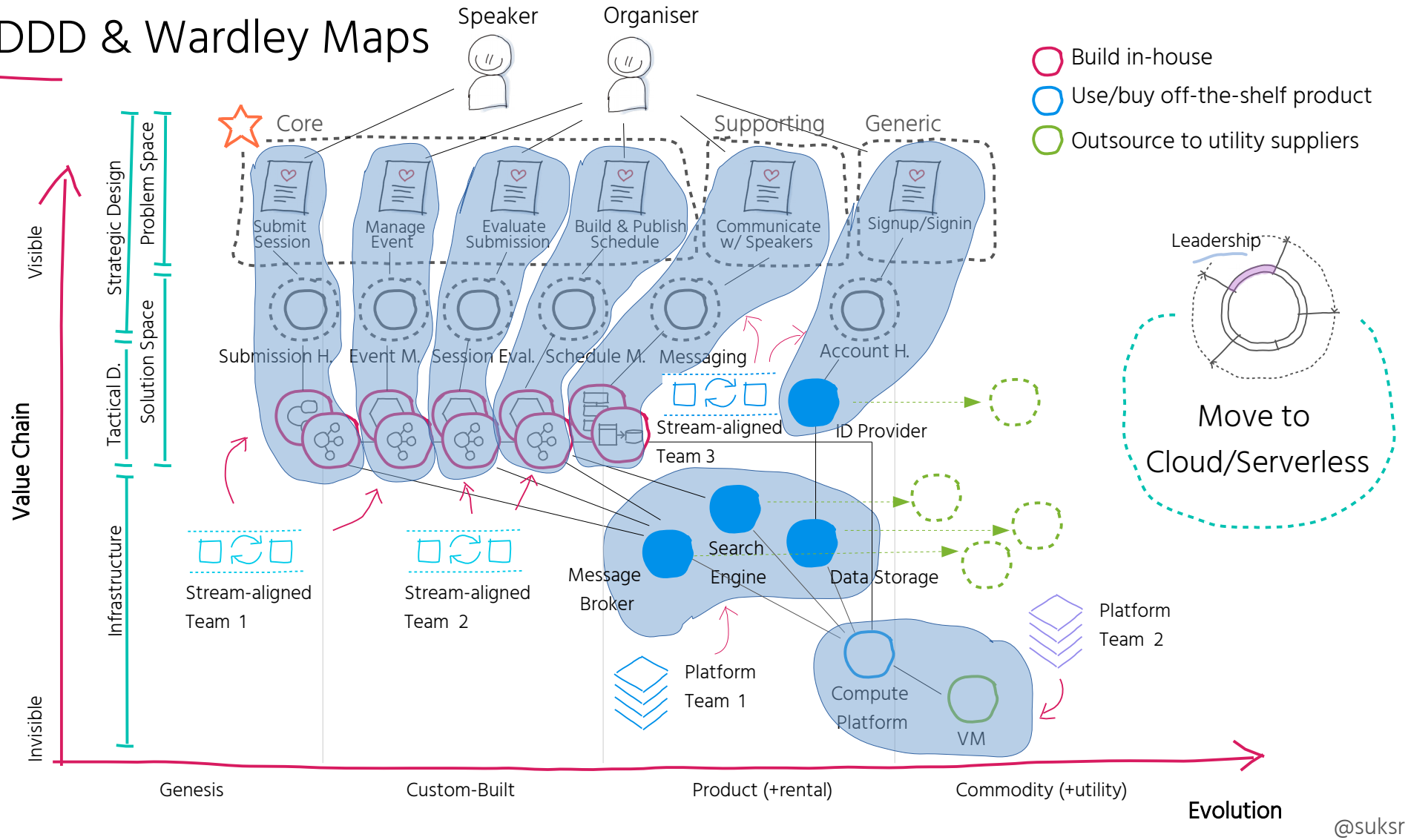
@suksr

TT & DDD & Wardley Maps



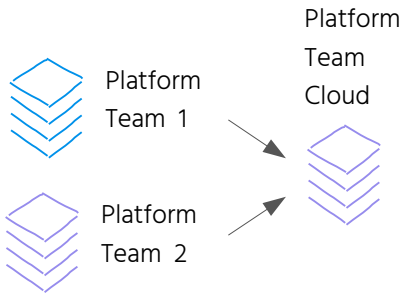
@suksr

TT & DDD & Wardley Maps

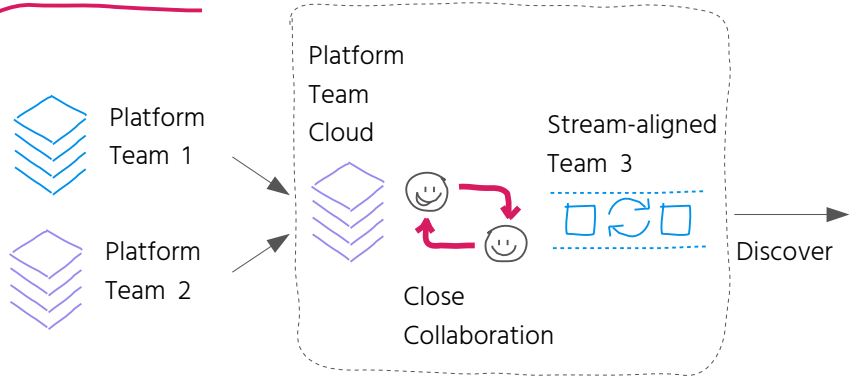


@suksr

Possible Evolution of Team Topologies

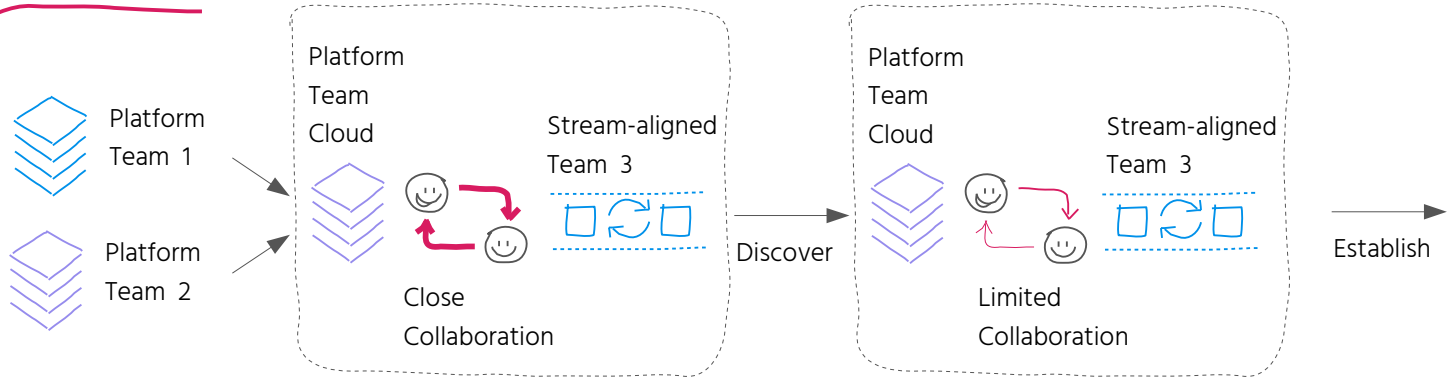


Possible Evolution of Team Topologies



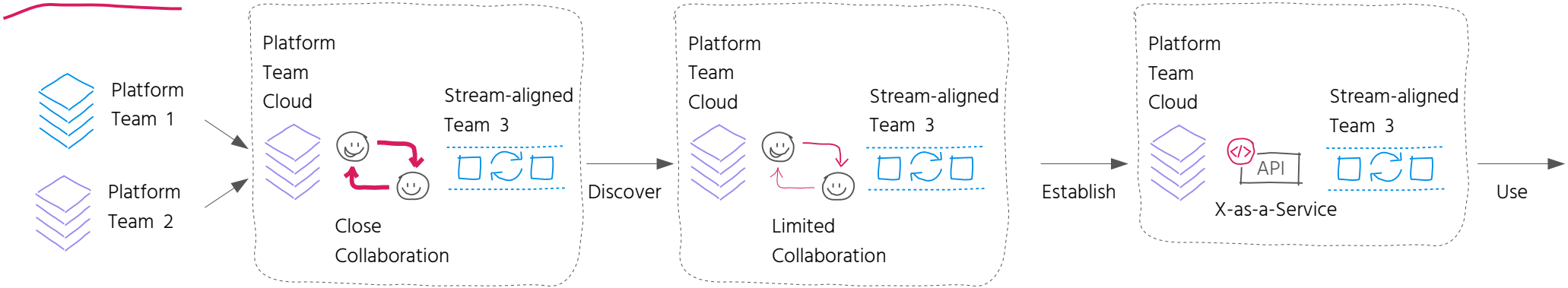
@suksr

Possible Evolution of Team Topologies



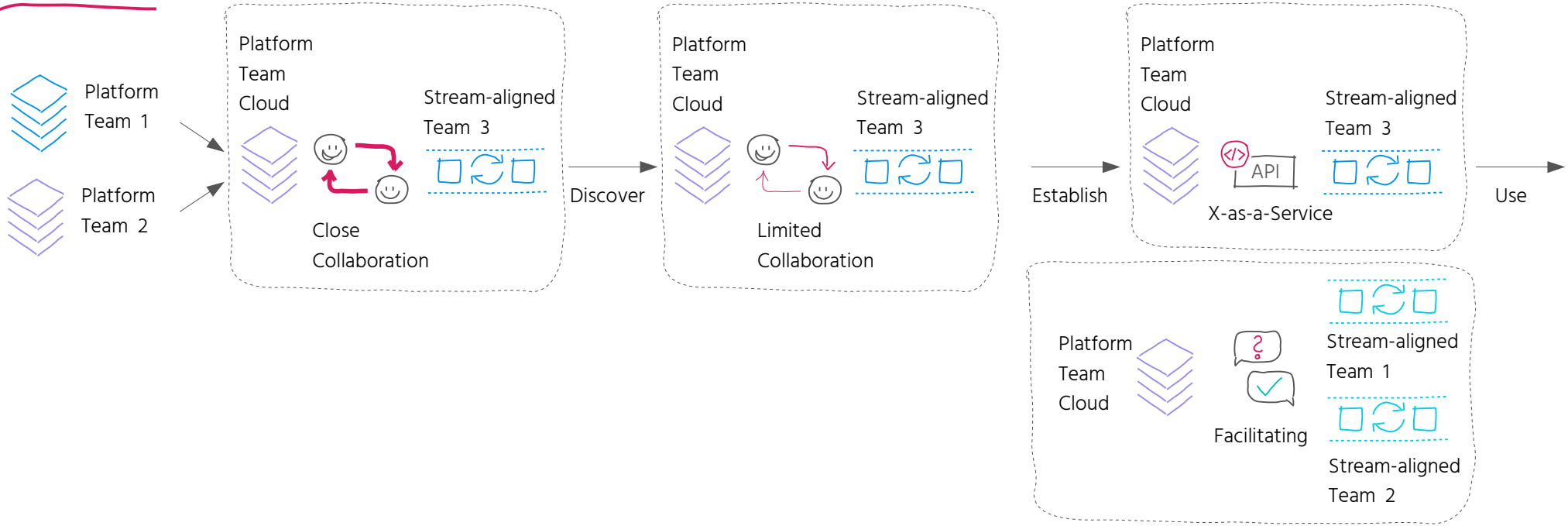
@suksr

Possible Evolution of Team Topologies



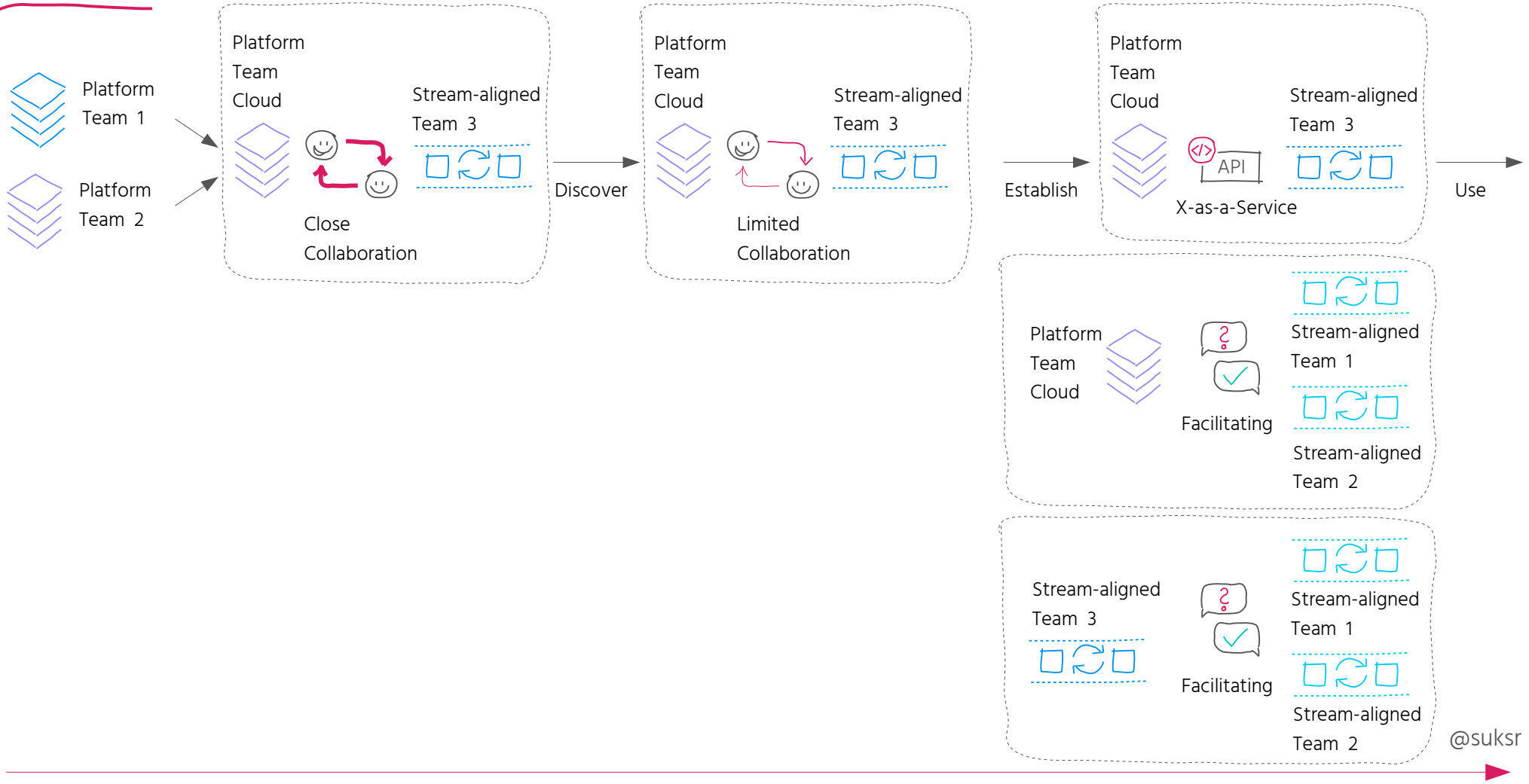
@suksr

Possible Evolution of Team Topologies

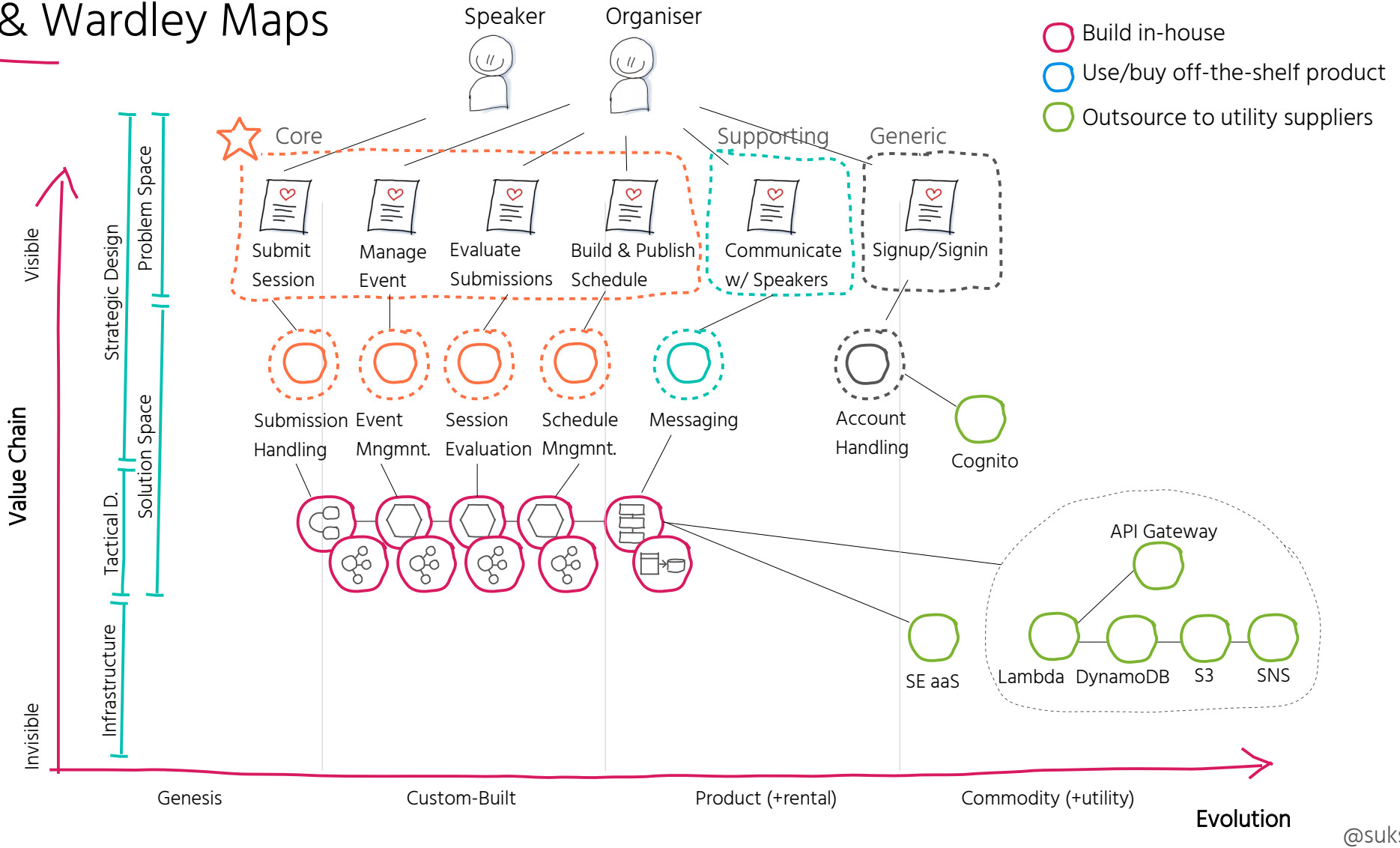


@suksr

Possible Evolution of Team Topologies

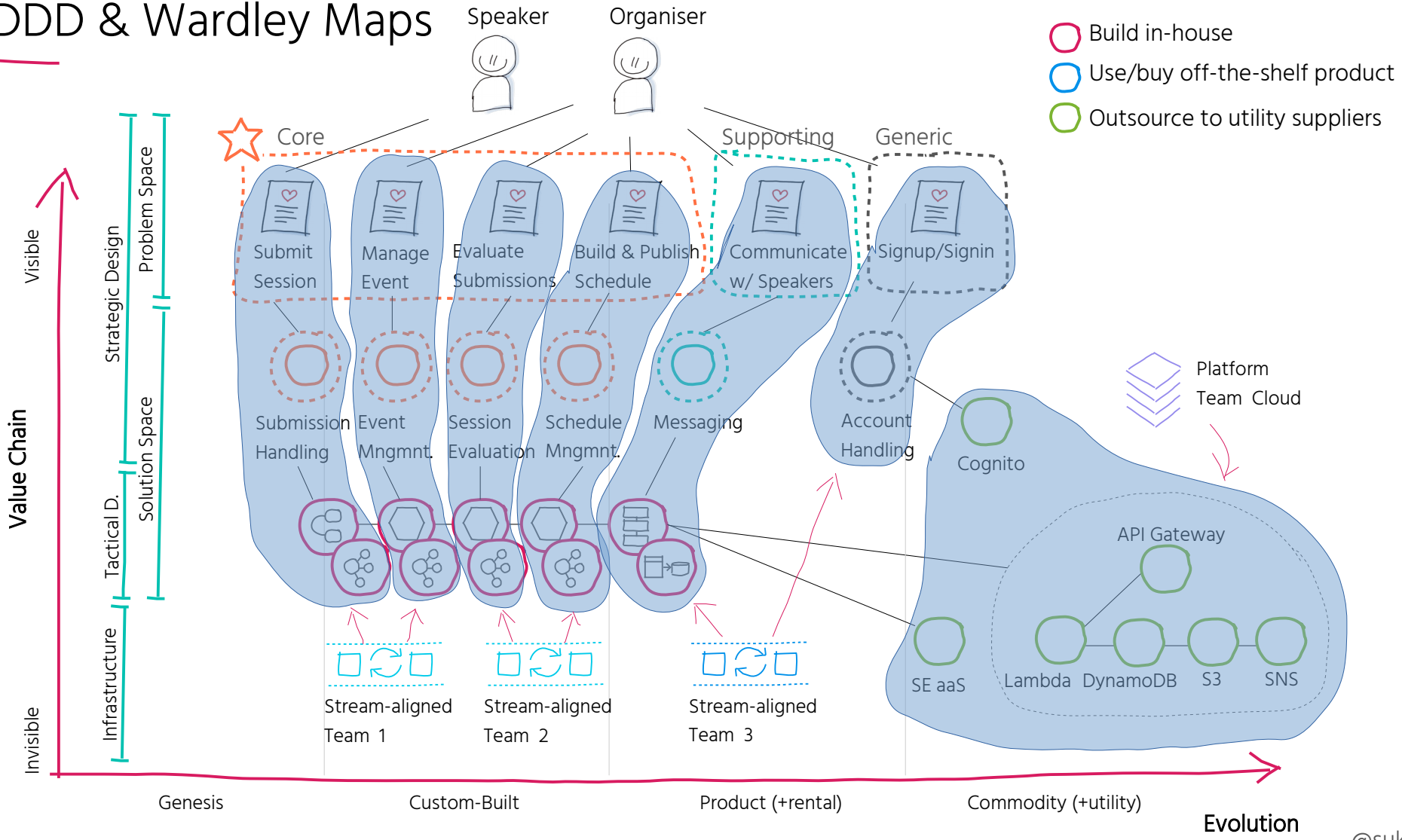


DDD & Wardley Maps



@suksr

TT & DDD & Wardley Maps

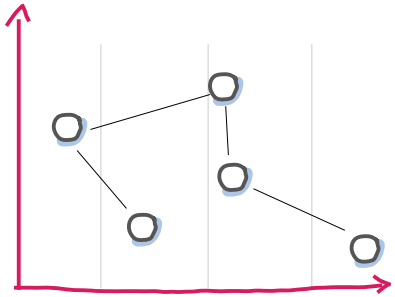


@suksr

Building Adaptive Systems for a Fast Flow of Change

1.

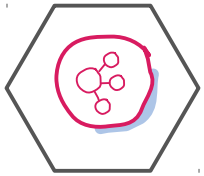
Business-Strategy



w/ Wardley Maps

2.

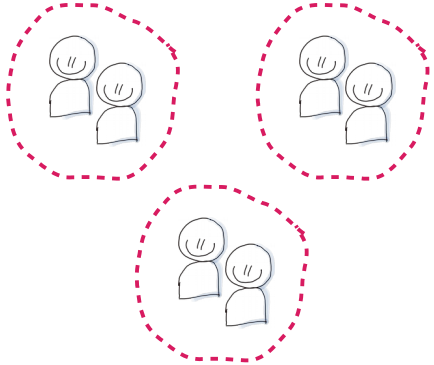
Software-Design/
-Architecture



w/ Domain-Driven Design

3.

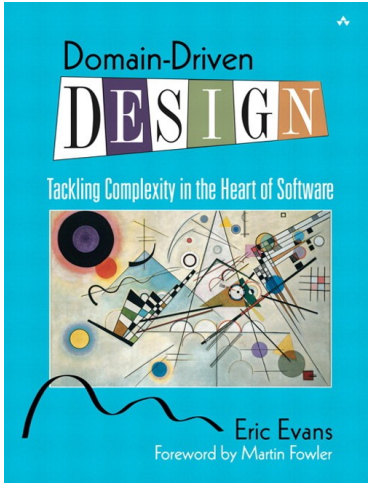
Team-Organization

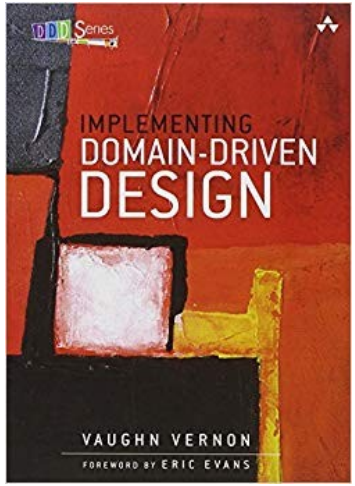


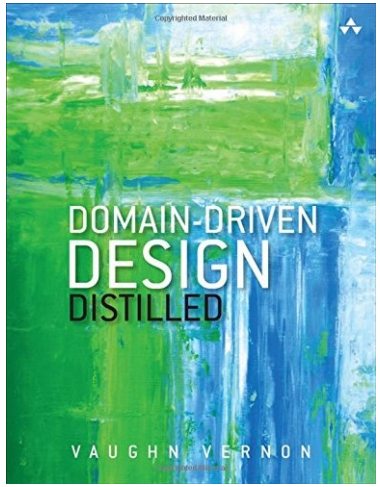
w/ Team Topologies

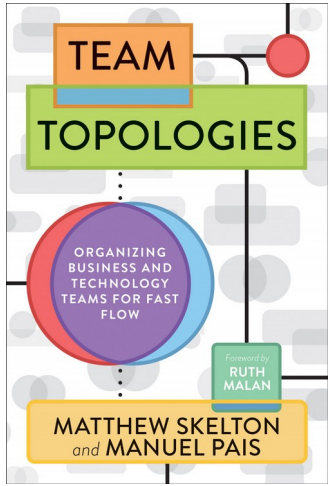
@suksr

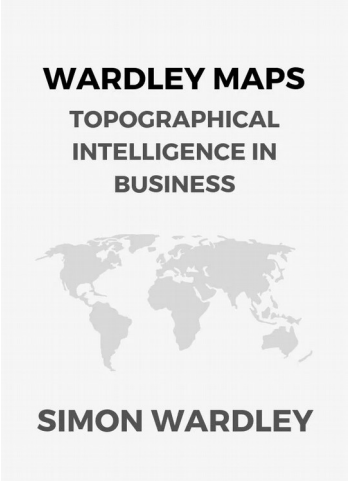
Some References











https://medium.com/wardleymaps
https://learnwardleymapping.com/
https://github.com/wardley-maps-community/awesome-wardley-maps

THANK YOU

Susanne Kaiser
Independent Tech Consultant
@suksr
susanne@susannekaiser.net