



**Systemic Framework for
Enterprise Architecture & Transformation**

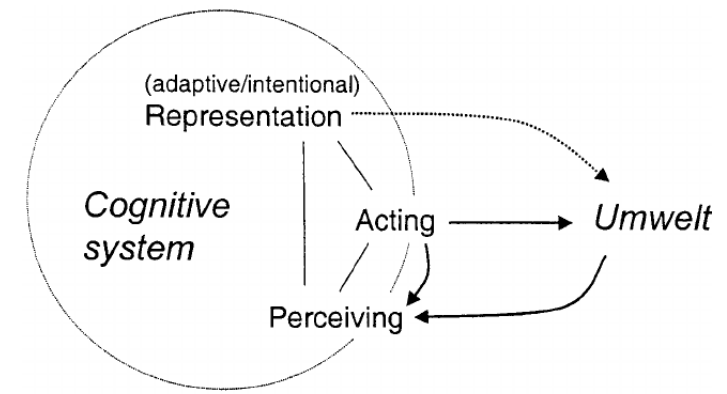
System Architecting

Models & Reality

The map is not the
territory

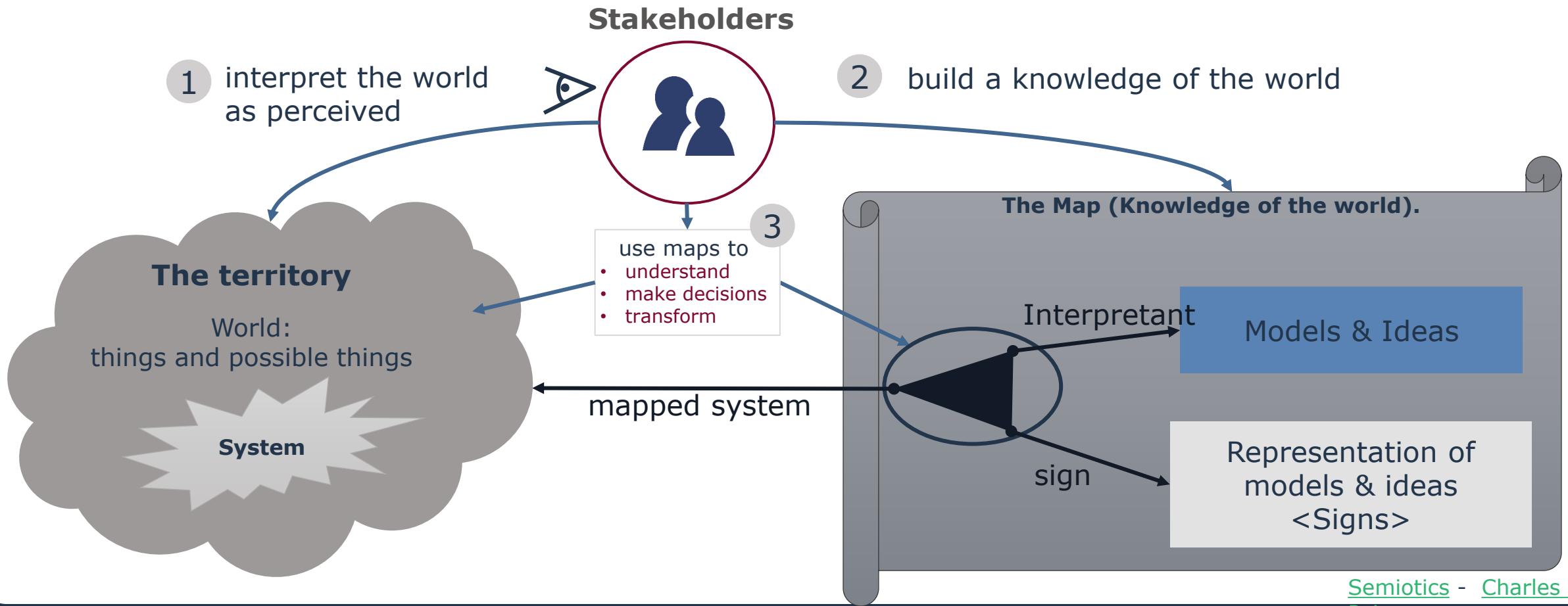
Reality, models and maps

- The description of the architecture of a system is an **abstraction**, a map:
 - *The map is not the territory (Alfred Korzybski).*
https://en.wikipedia.org/wiki/Alfred_Korzybski
- Model construction is part of a semiotic process:
 - Models are **constructed** representations of the enterprise, by the system in its environment.
 - purpose-based understand itself in its environment,
 - act based on this understanding,
 - adapt and change itself based of feedbacks,
 - ... all this over time through experience and learning.
- The semiotic process relies on a **purpose-based representation** (theological) of the system in its environment.



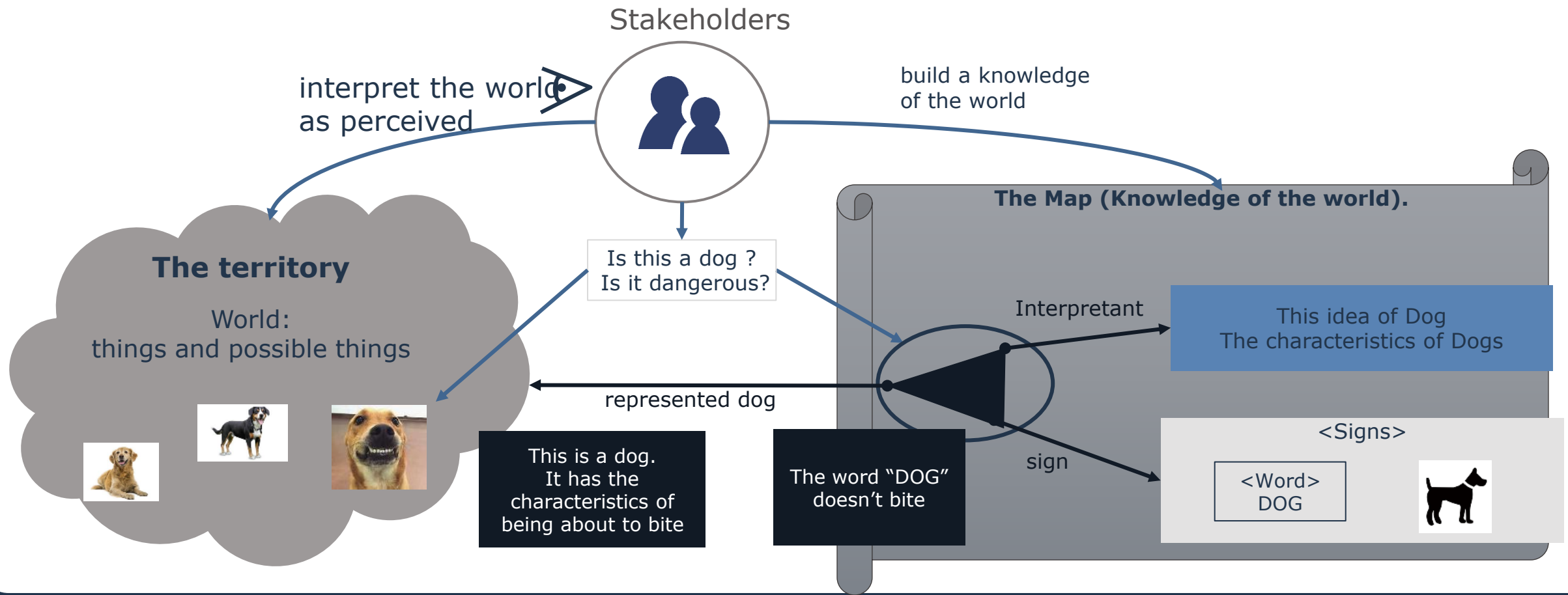
The map, the territory and stakeholders

- The map is built by people, who have some vested interests in a **joint undertaking**, where the map is used for **understanding**, **decision making** and **transformations**.
- Each object in the map is a triangle, between the denoted thing or class of things in the world, the idea about that thing and the signs or symbols used to represent the idea (semiotic).



We need maps of the world, remembering the map is not the world

- Without maps it is not possible to make statements about the world:
 - For instance, is the above animal a dog ? Is it dangerous? What should I do based on my understanding?
- Feedback loops with reality are always necessary to ensure the accuracy of models:
 - The word "Dog" doesn't bite. Only our experience of dog bites will validate the accuracy of the model.



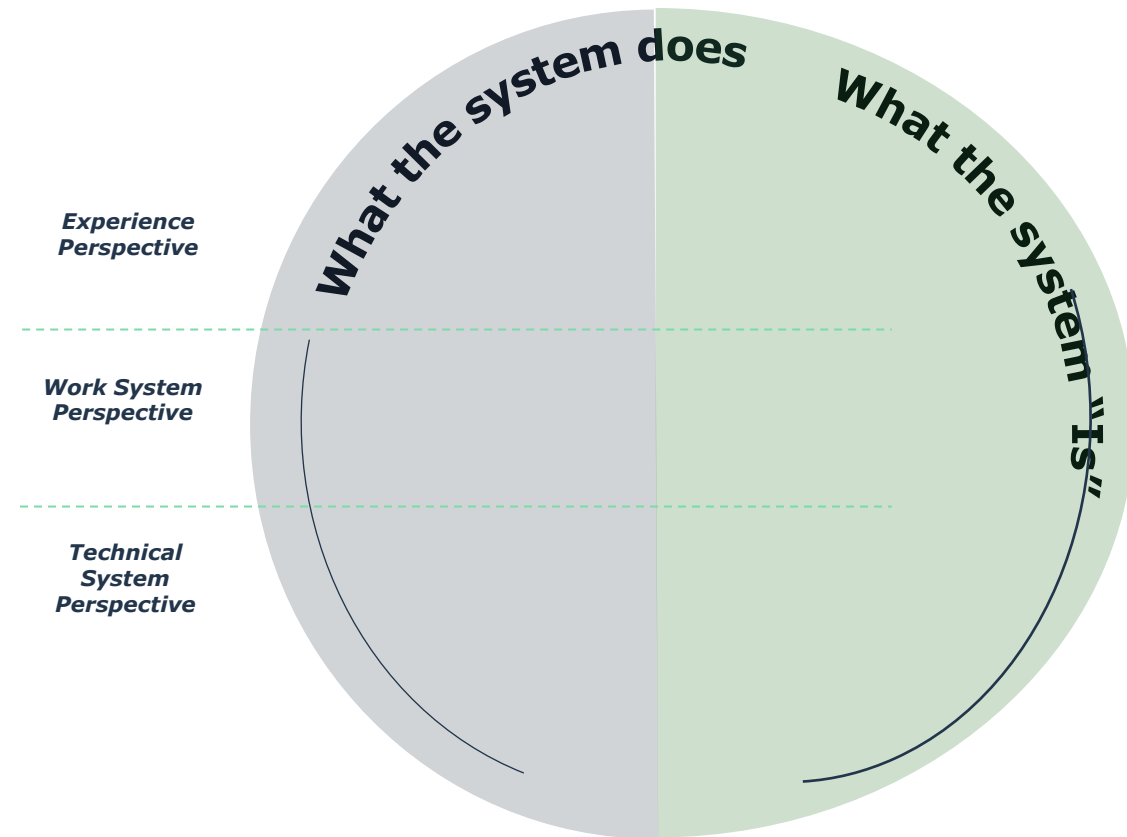
System Architecting Method

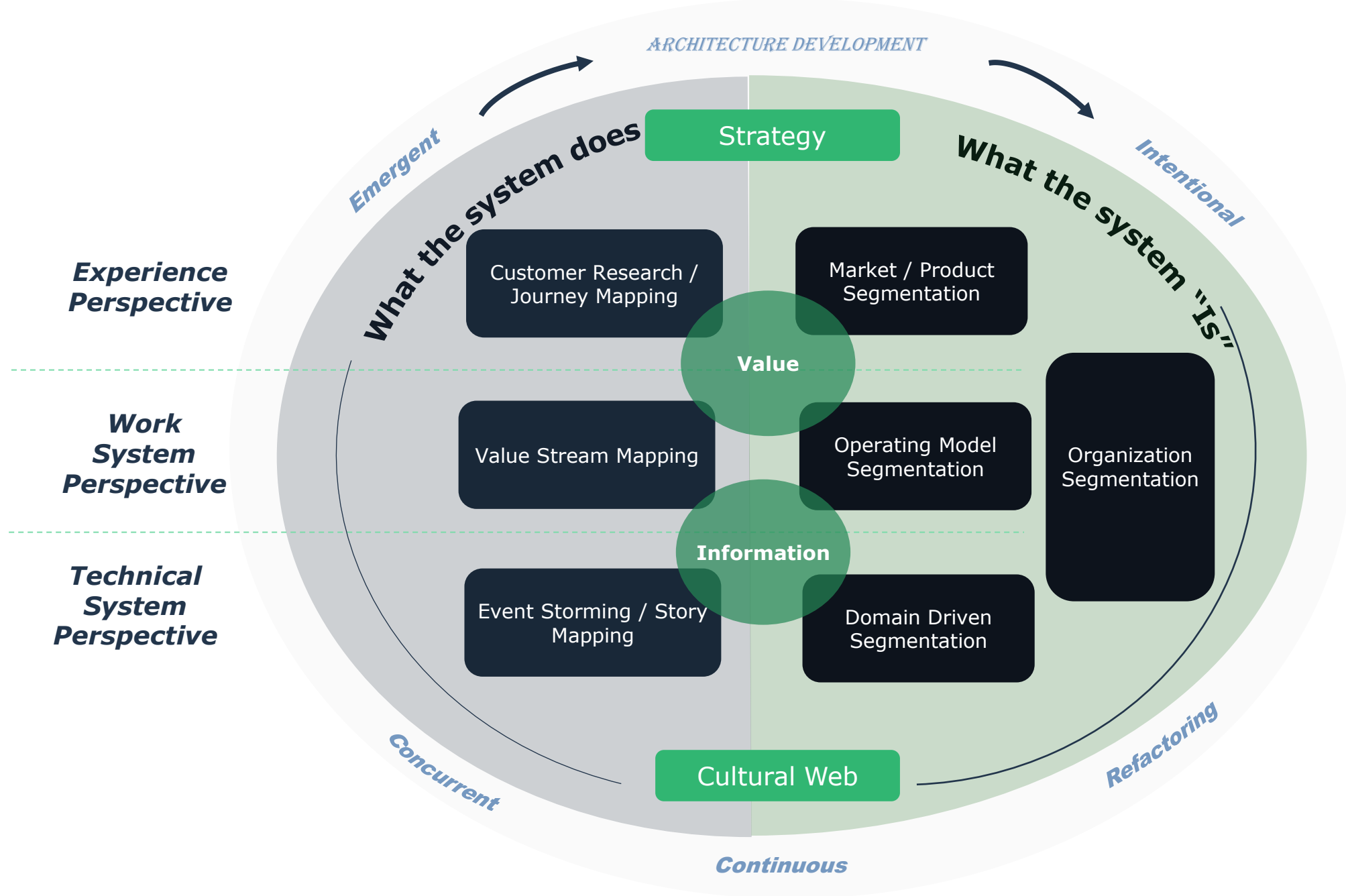
What is it?

What is it used for?

How to build it?

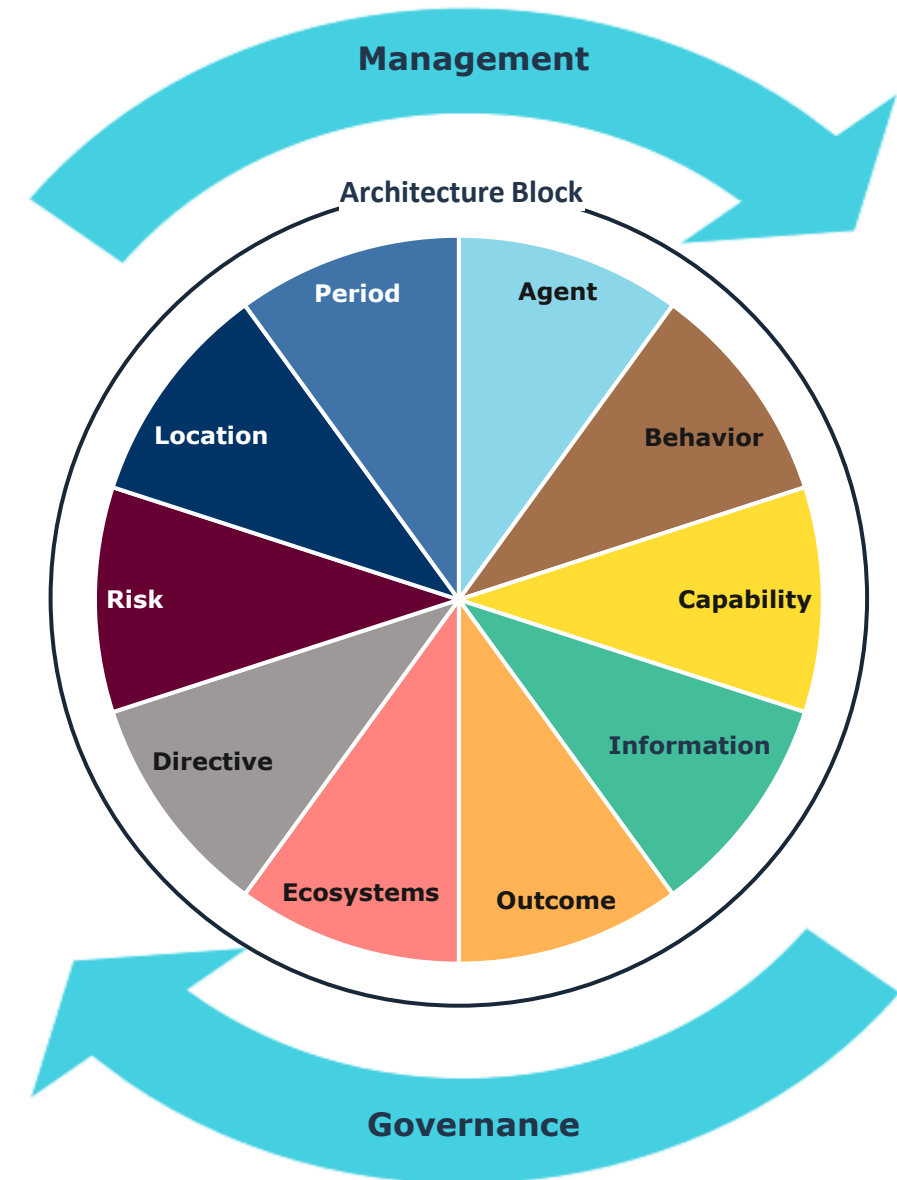
- Traditional division between “What the system does” and “what the system is”

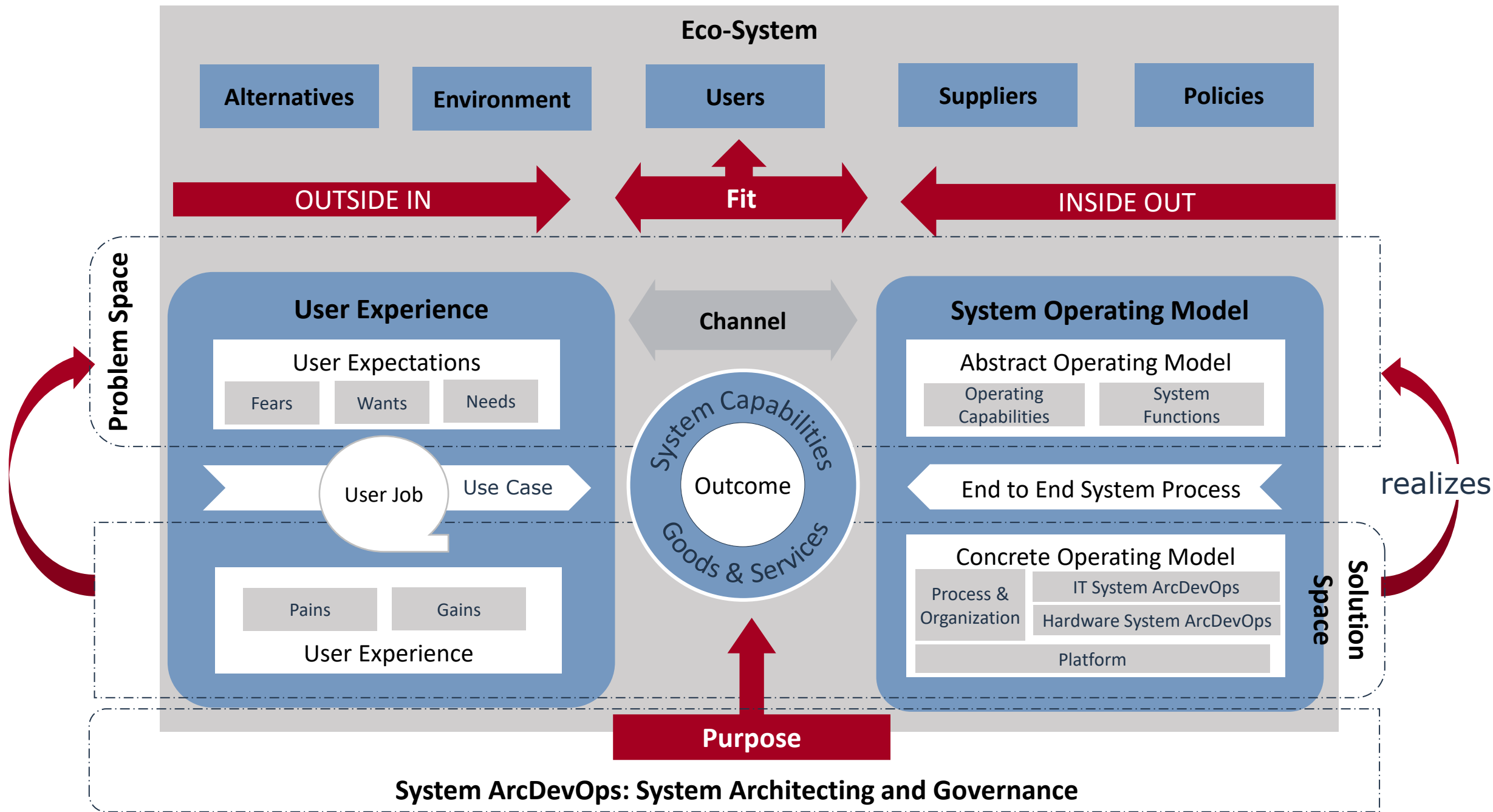




Outcome driven architecting

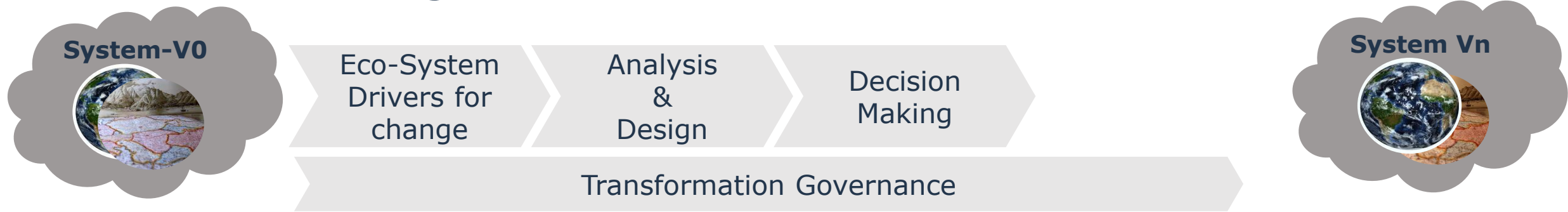
- The various aspects that explain why and how a system “functions” are outlined in the diagram on the right.
- A key aspect of the method is the distinction between:
 - User journeys which express user outcome expectations in their job context.
 - Capabilities which express abilities of the system to produce expected outcomes.
 - Operating Models which express the way agents operate to deliver expected outcomes.
- All are positioned in an Eco-System where:
 - Capabilities are offered in regard to user jobs (outside-in).
 - Operating solutions are experienced by users doing their jobs (inside-out).
 - Stakeholder drivers and goals express a volition to respond to ecosystem expectations and experience, overtime.





- Avec les propriétés
- Faire tel « travail », à telle vitesse
- Les pistes ont telle longueur: $\Rightarrow$ a des conséquences sur la propriété accélération $\Rightarrow$ c_x pénétration dans l'air sous forme de x et y est la portance.
- $\Rightarrow$ problème de traçabilité des propriétés entre conceptuel et organique, tout en remontant à la capacité de départ (Transport)
- Les pistes sont caractérisées:
 - les features
 - Il faut parler de l'atmosphère en fonction de l'altitude
 - On avance parce que l'on s'appuie sur quelque chose plus lourd que soi
 - Idem, il faut que l'air ait de la portance.
 - Comment caractériser l'air et donc ses propriétés. Comment différencier l'air
- Aérodynamique : c'est la relation entre une forme en mouvement dans un milieu gazeux.

The architecting method

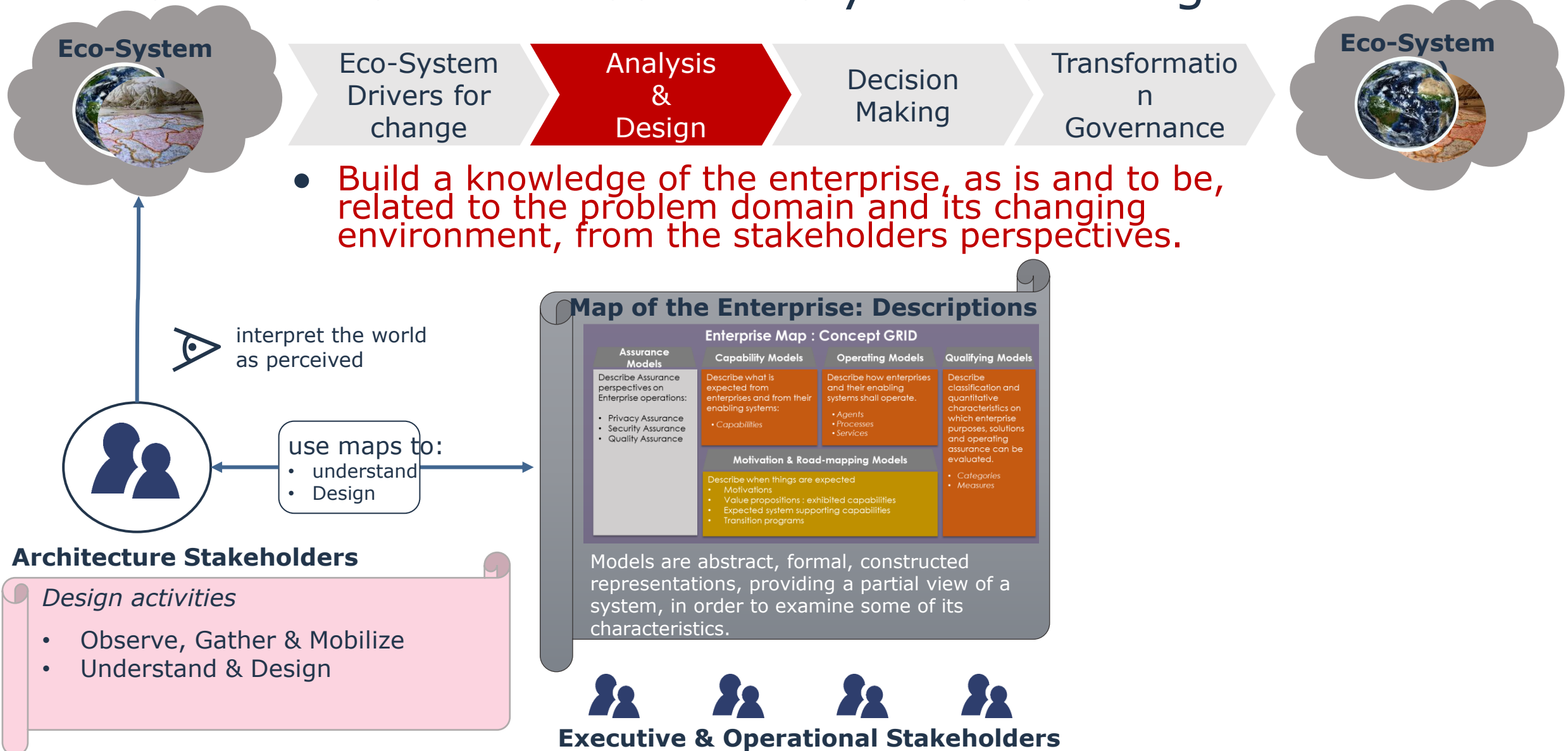


- Provide continuous architecture through:
 - Embedding the System in its Eco-System.
 - Managing Stakeholder involvement.
 - Delivering an operating model align with a usage model.
 - Recording functional and architecture decisions.
 - Federating distributed autonomous teams through monitored feedback loops.

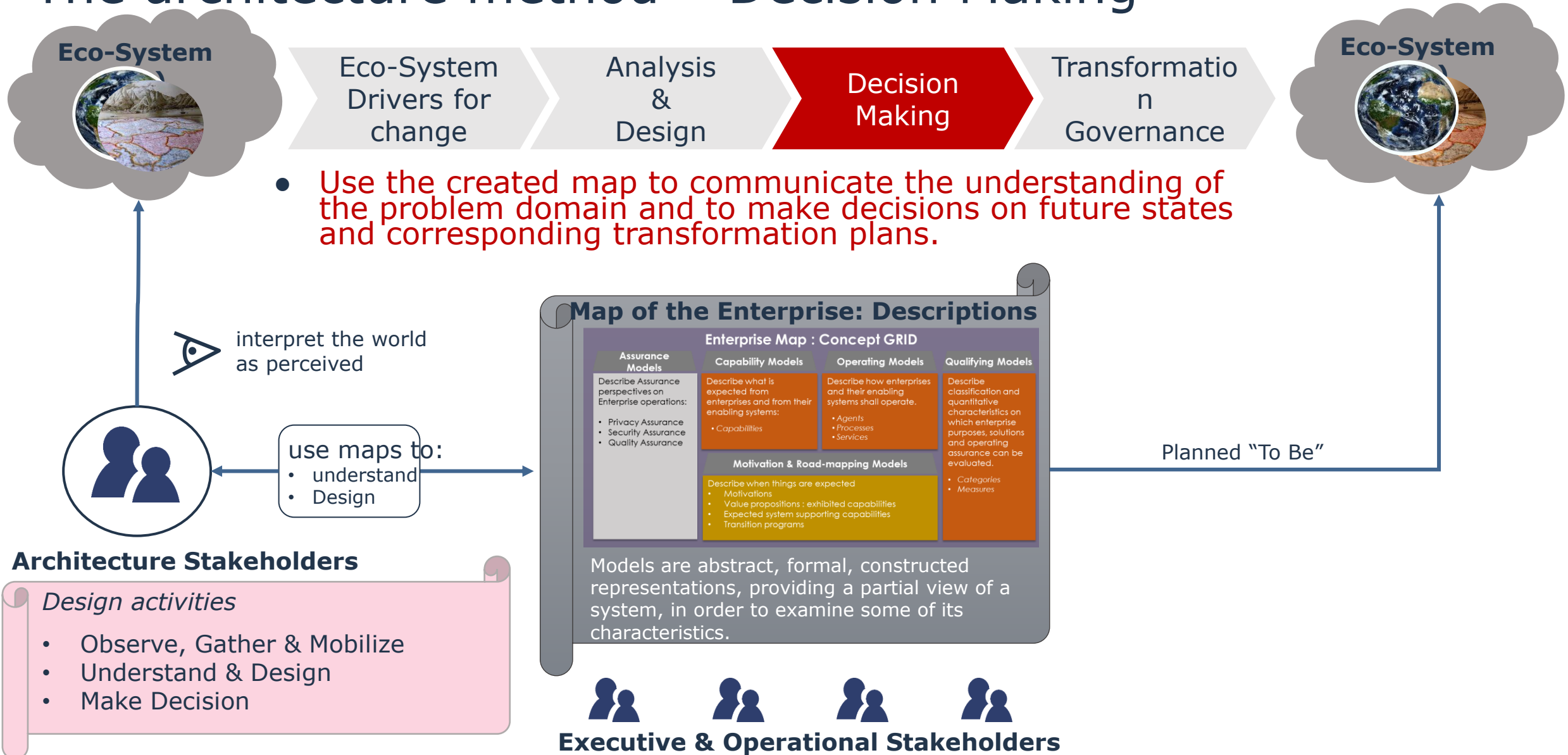
The architecture method – Inception and Scoping



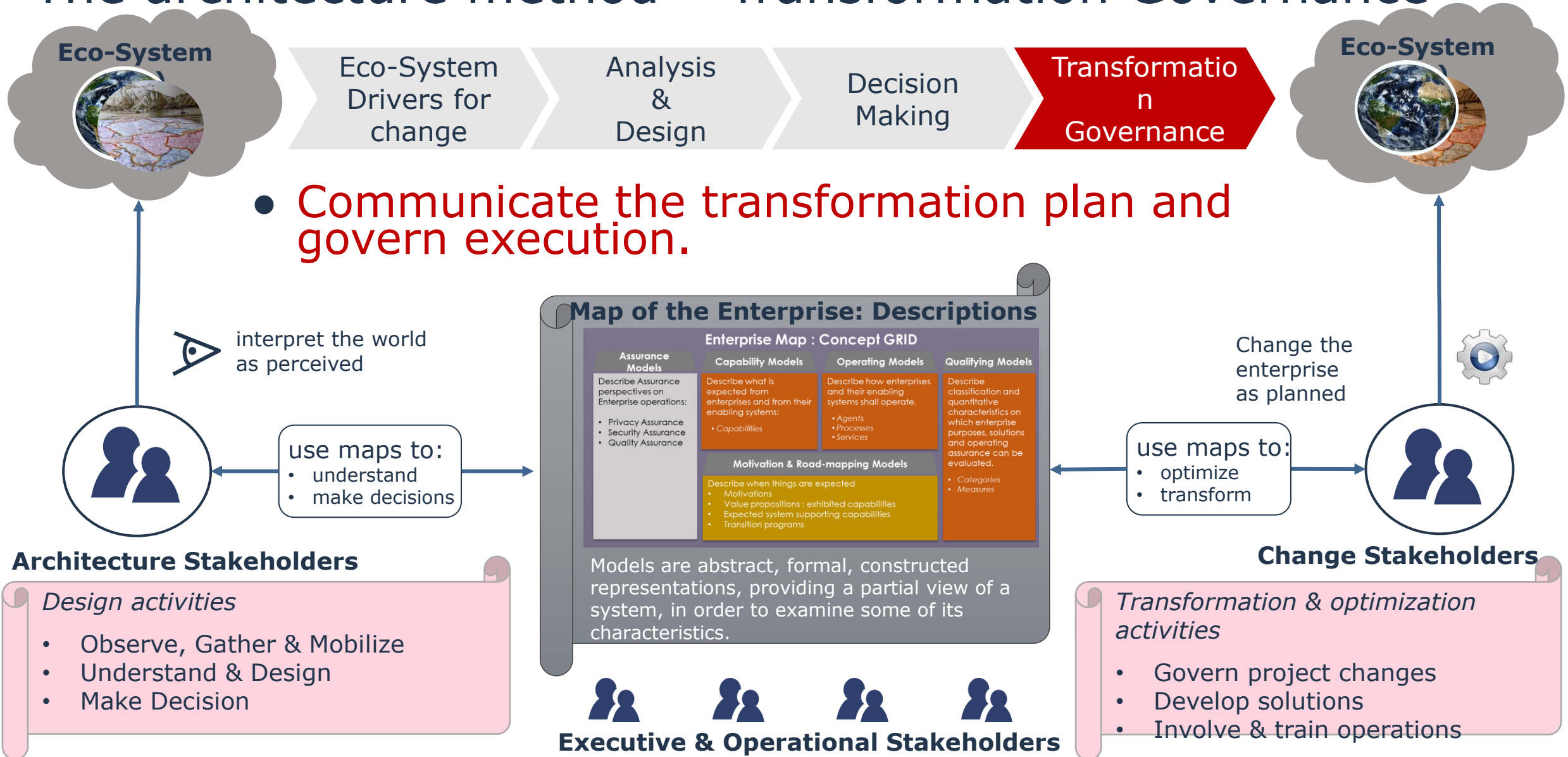
The architecture method – Analysis and Design



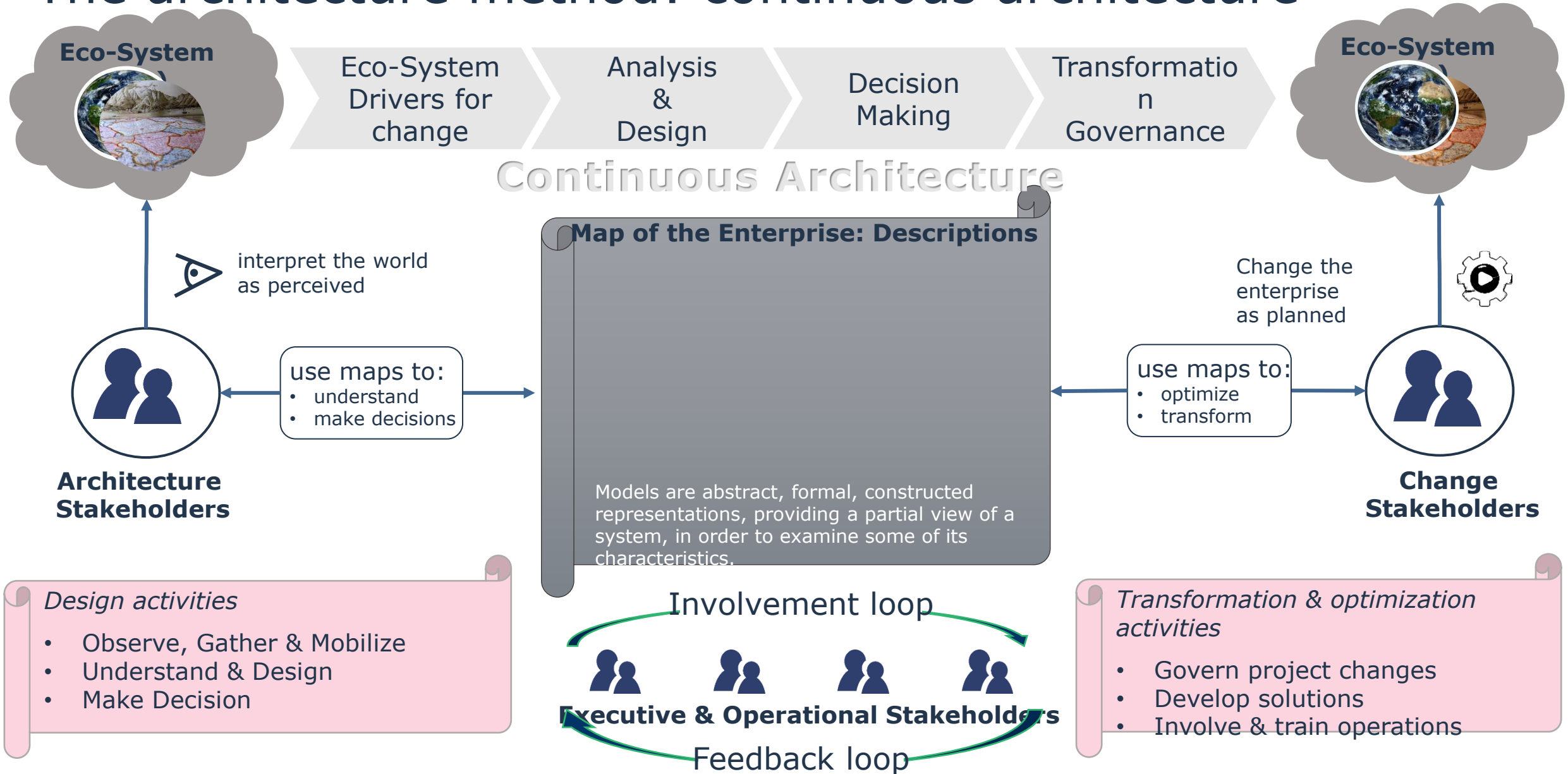
The architecture method – Decision Making



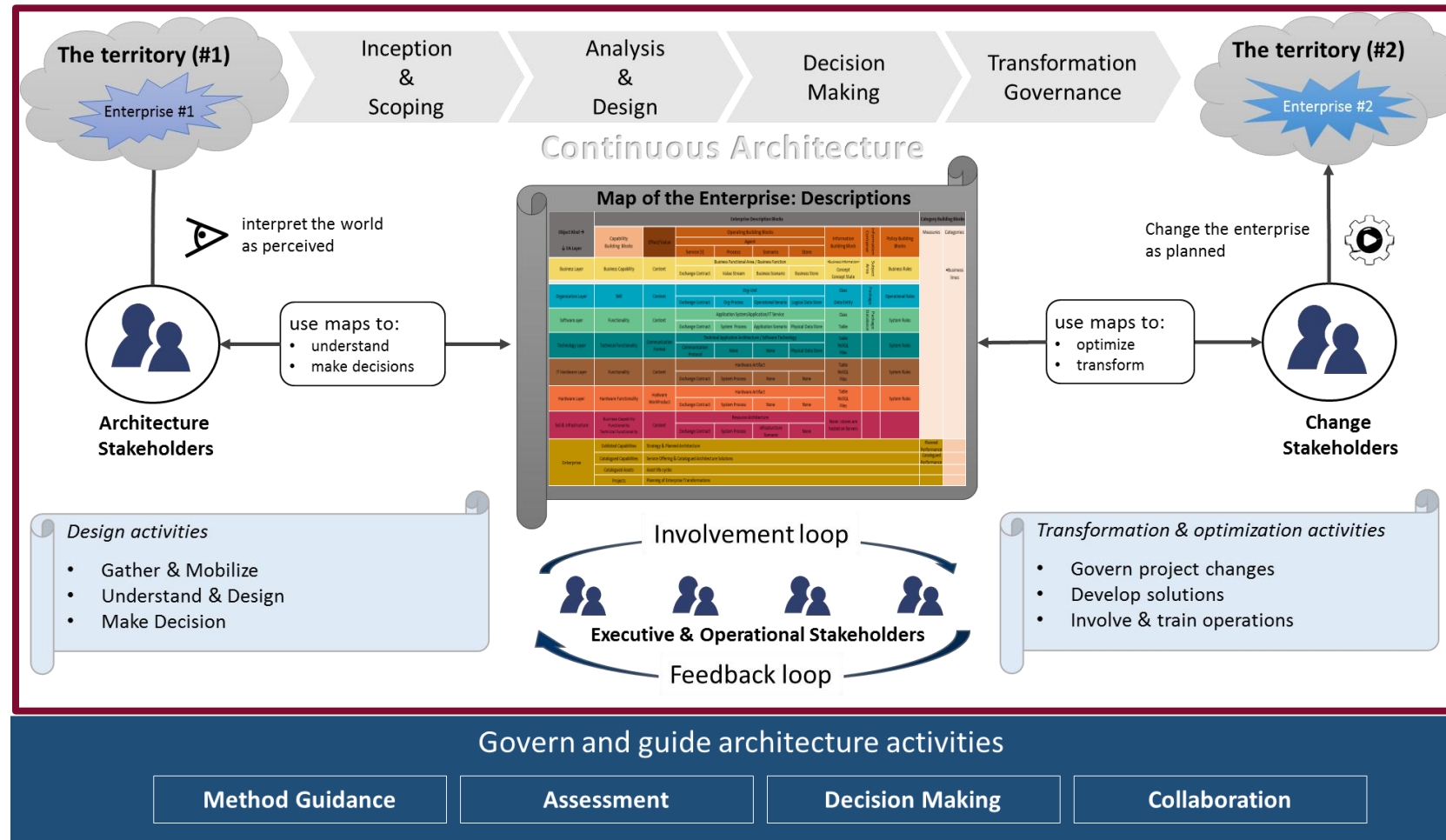
The architecture method – Transformation Governance



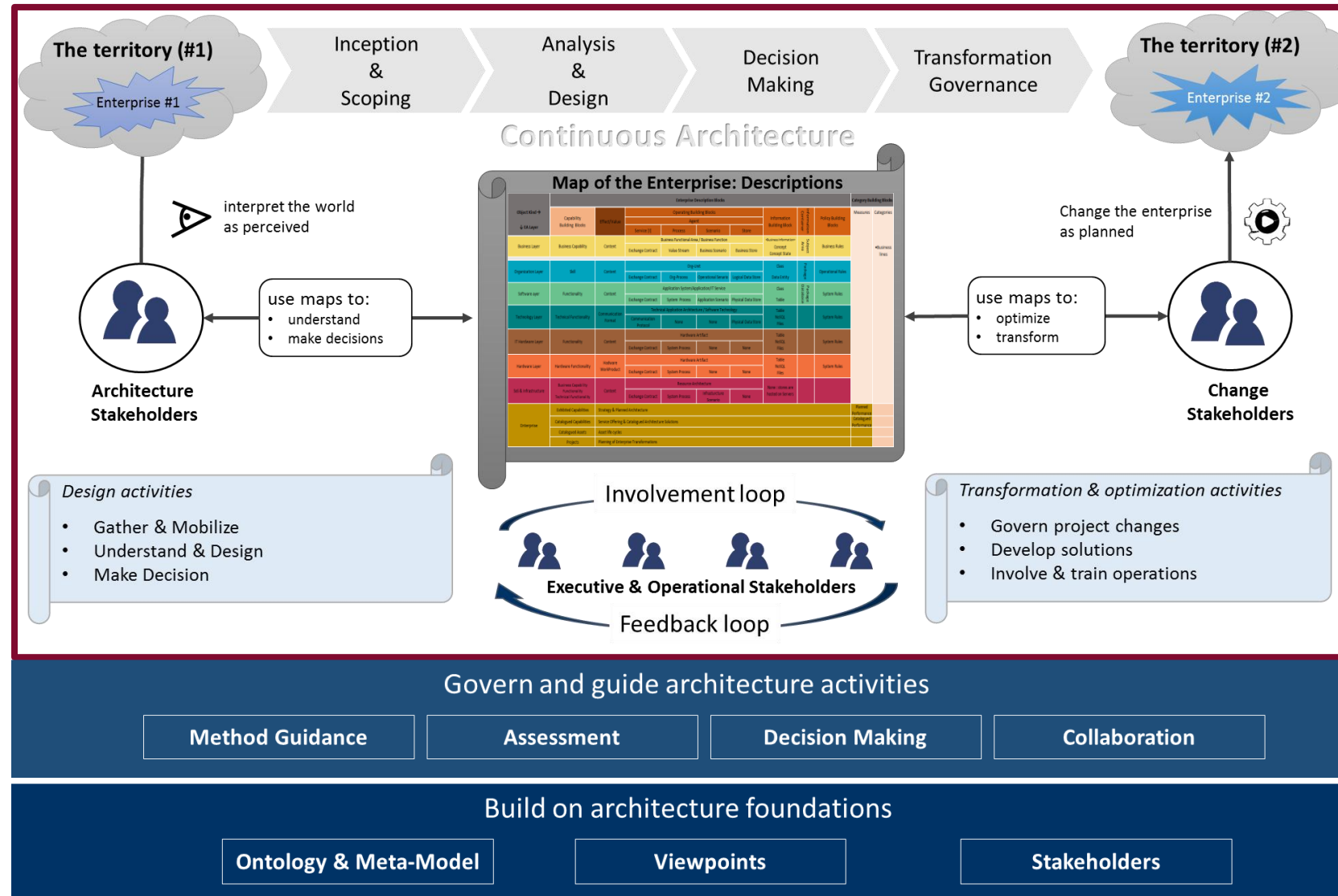
The architecture method: continuous architecture



The enterprise architecture framework



The architecture framework

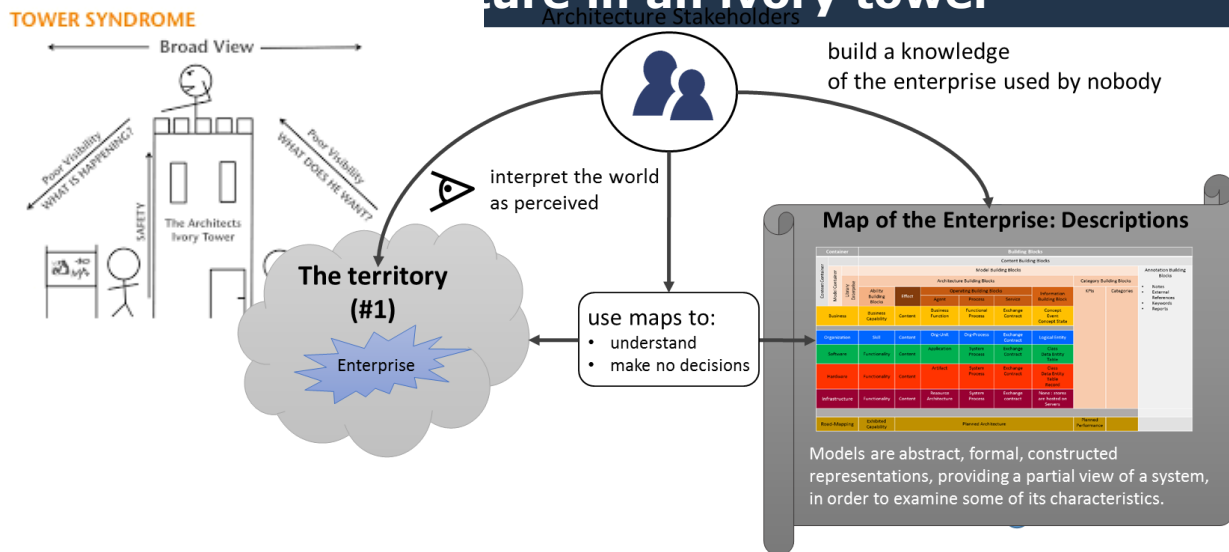


The modeling method benefits

- Architecture Descriptions without experience and implementation are mere theory.
- Implementations without architecture guidance ends up in building garbage and slums.
- The Enterprise Architecture discipline allows to setup the right amount of architecture and experience work to nurture adaptable and sustainable businesses.

Architecture Description without practical feedbacks:

Architecture in an ivory tower



Implementation without guidance: tinkering The route for garbage and slums



System Operating Model

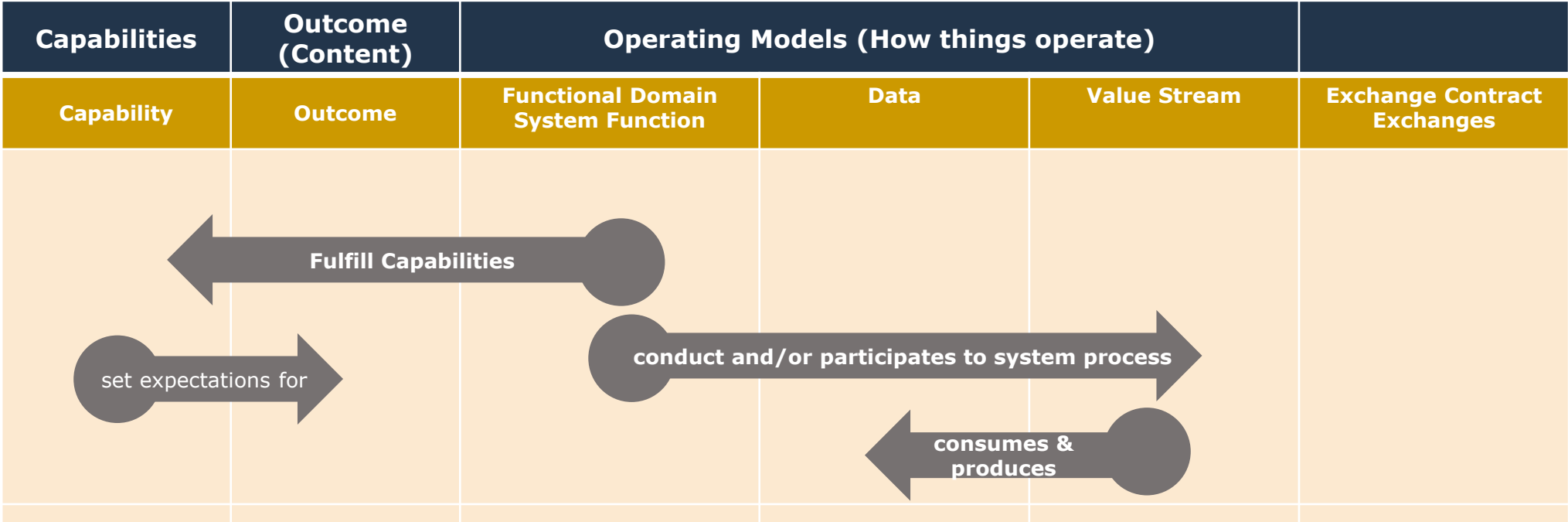
What is it?

What is it used for?

How to build it?

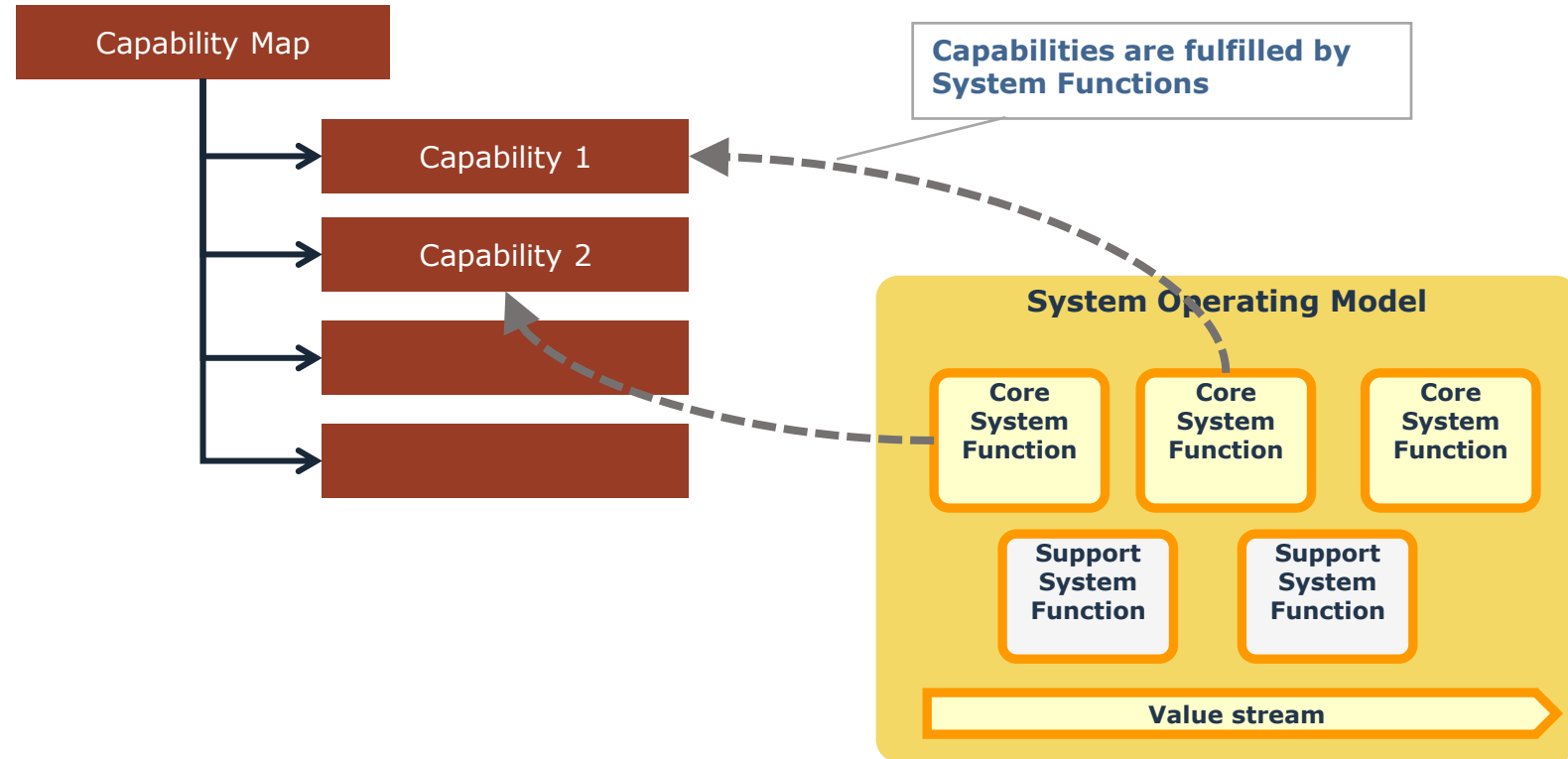
System Operating Model

- Capabilities: functionalities ability to produce and outcome.
 - Capabilities express what can be offered by Agents, without committing on how Agents behave.
- Outcomes: results, expected by Capabilities and delivered by Agents.
- Agents: who is doing, who is in charge.
- Interactions: service interactions between Agents.
- Actions: activities done by Agents (processes).



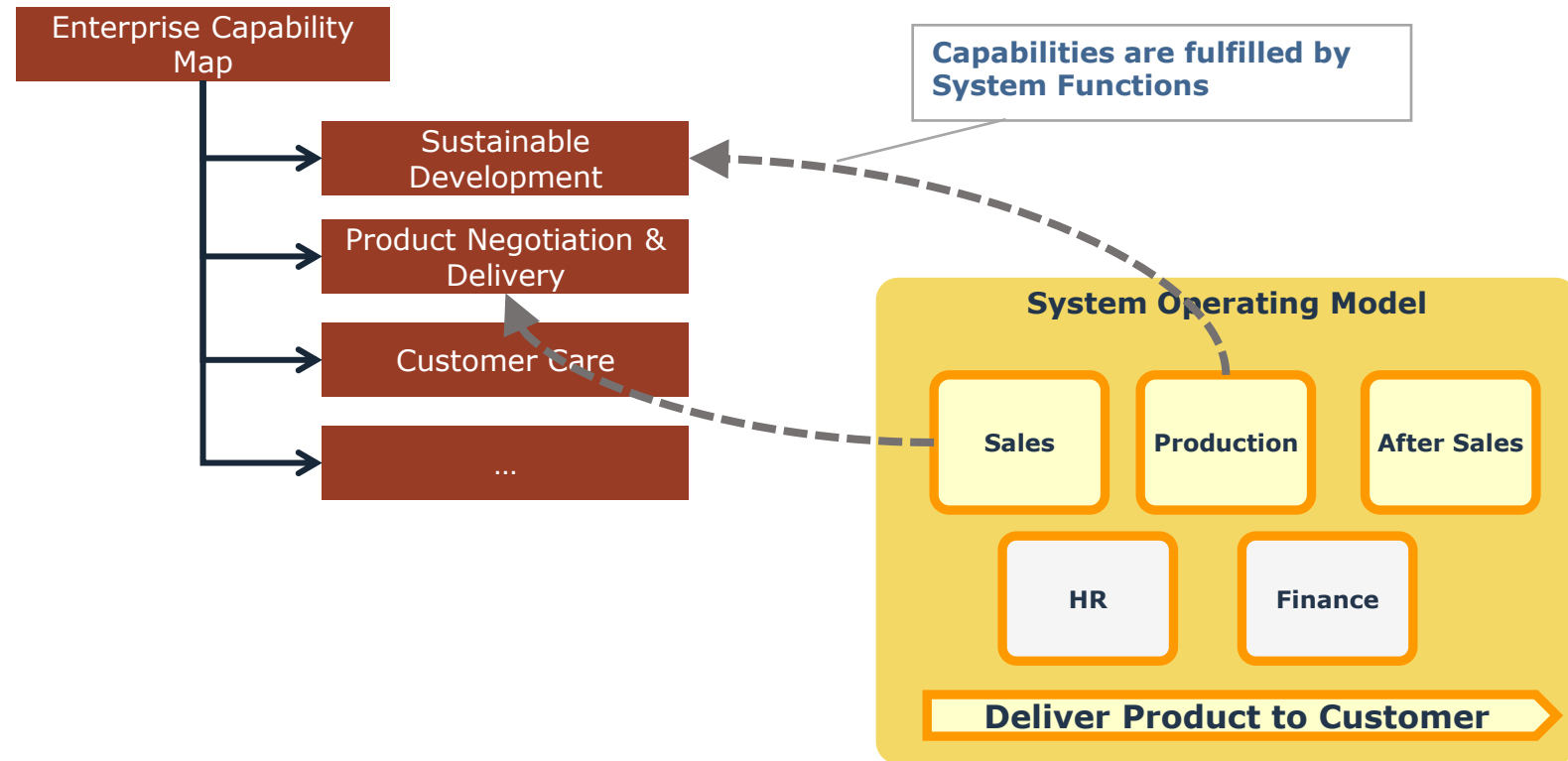
Strategic Alignment of System Operating Model

- Capabilities are fulfilled by System Functions.



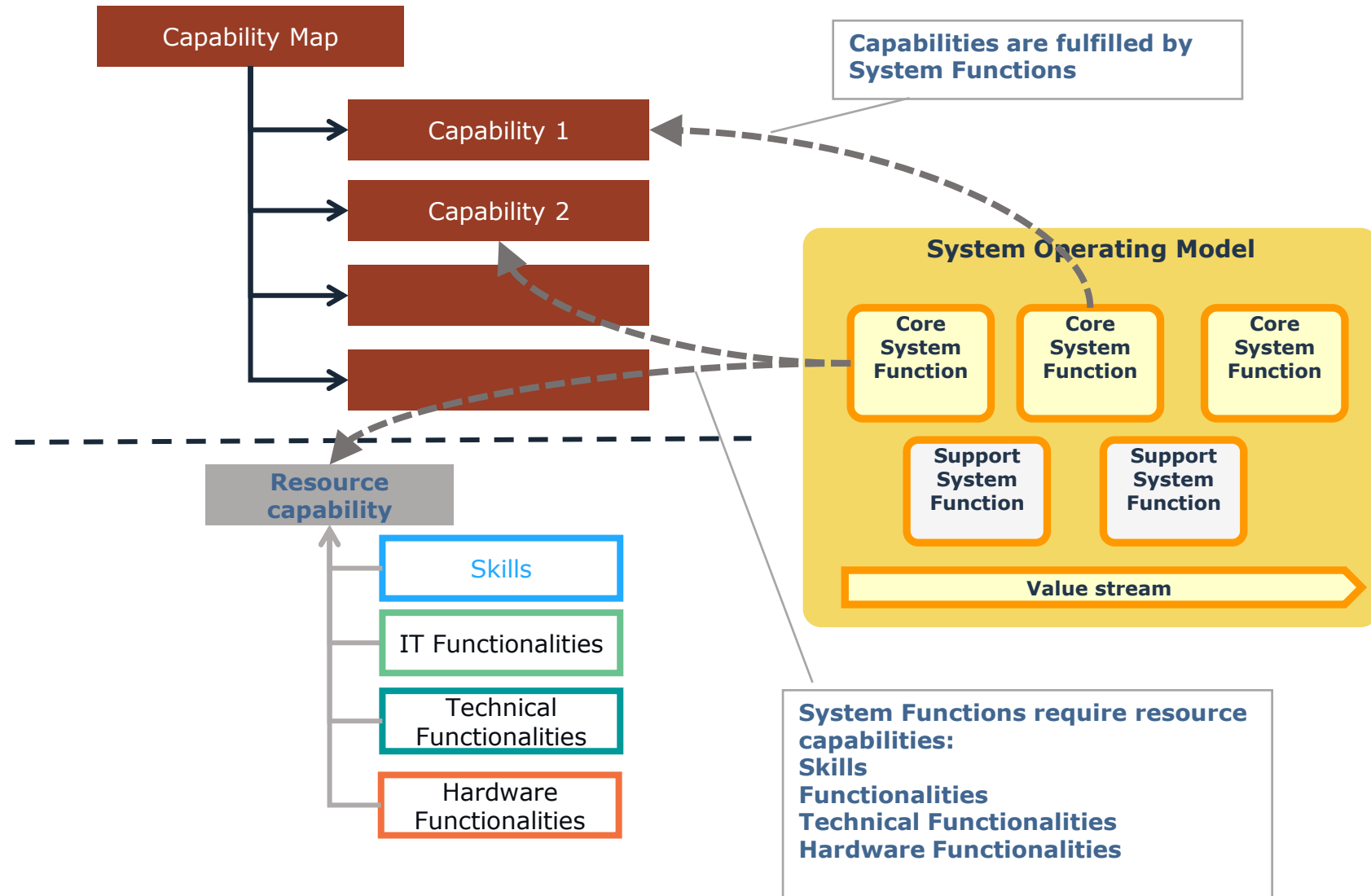
Strategic Alignment of System Operating Model

- Capabilities are fulfilled by System Functions.



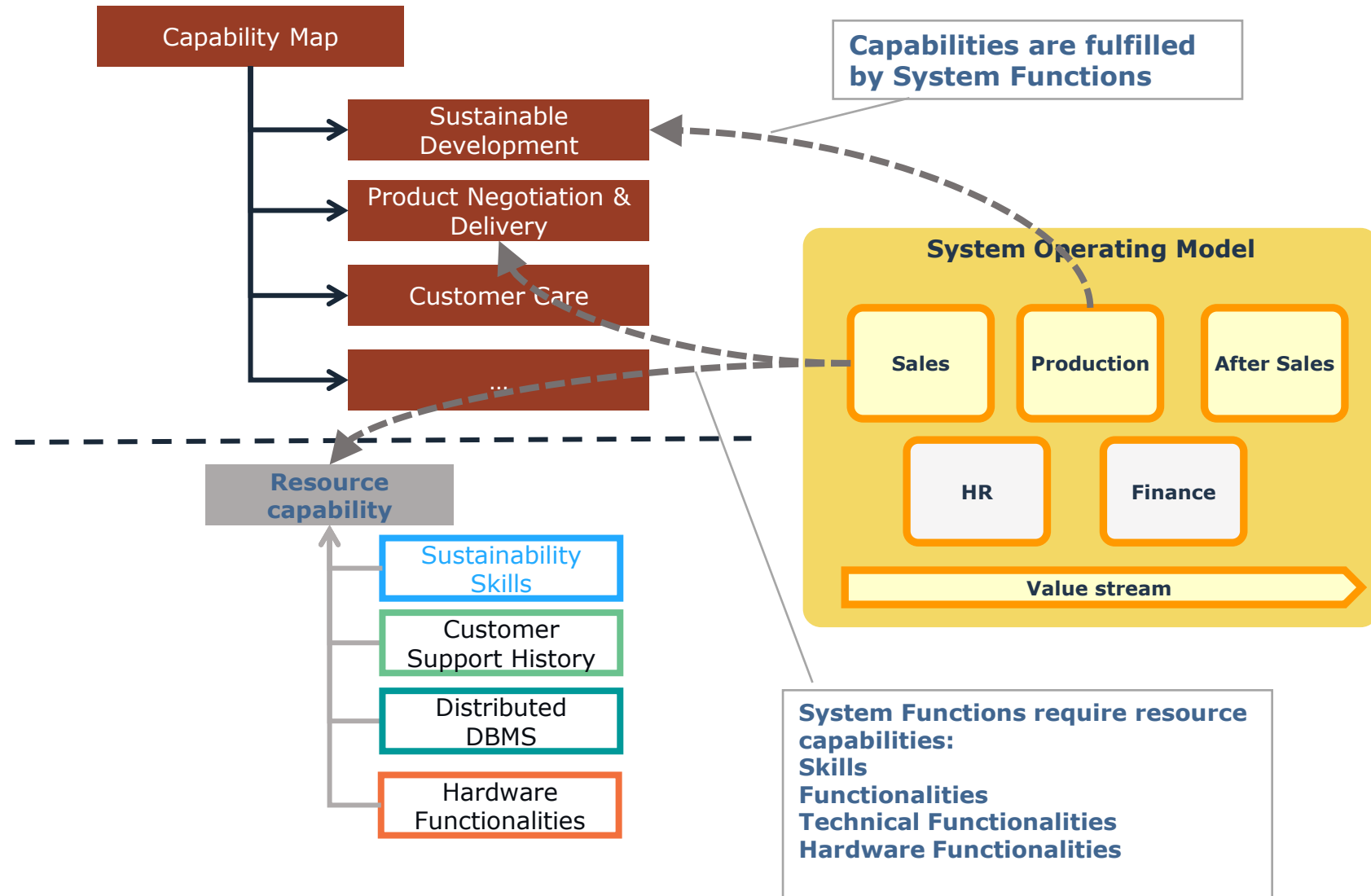
Strategic Alignment of Resources Capabilities

- Capabilities are fulfilled by System Functions.
- System Functions required Functionalities/Skills



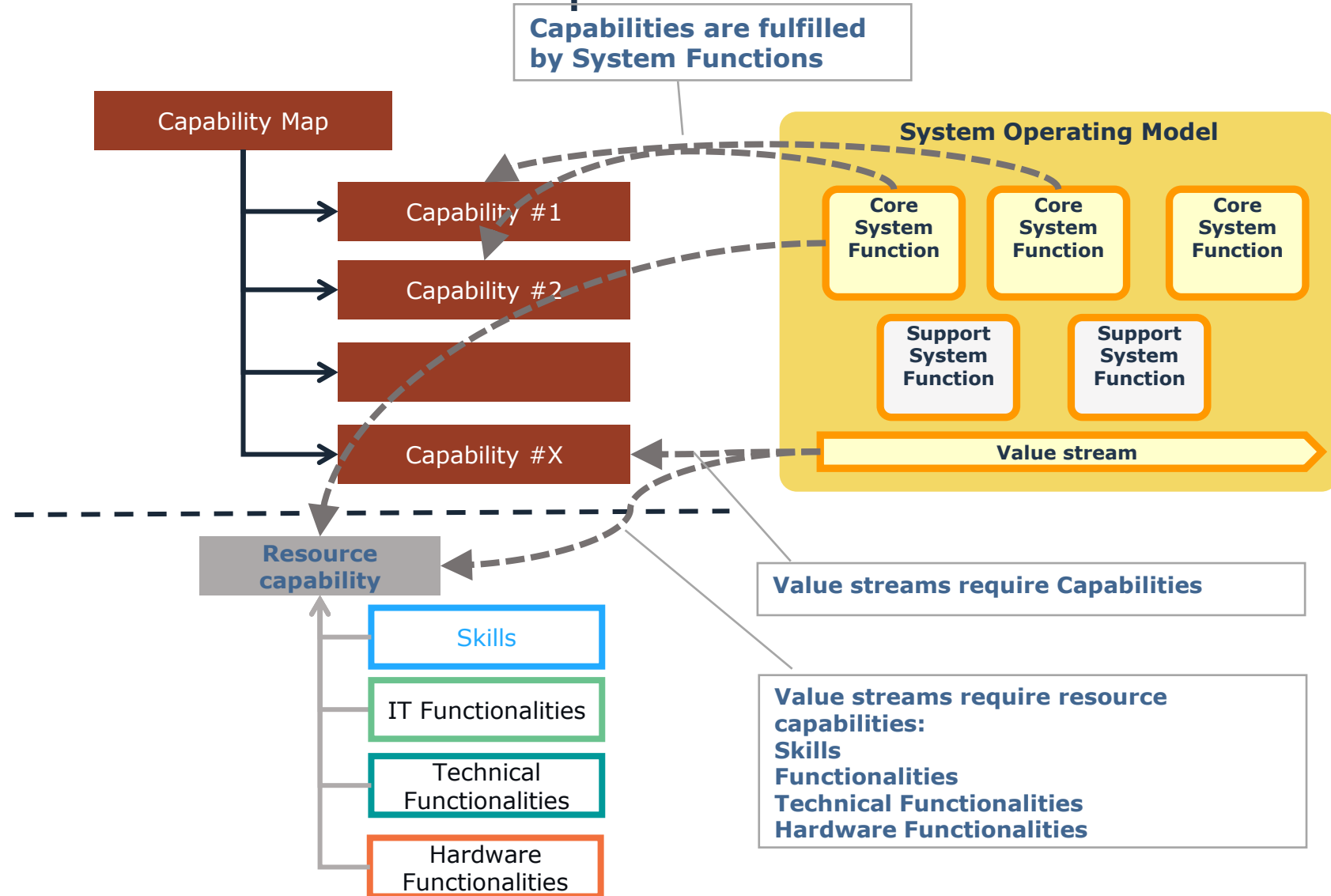
Strategic Alignment of Resources Capabilities

- Capabilities are fulfilled by System Functions.
- System Functions required Functionalities/Skills



Strategic Alignment of Resources Capabilities

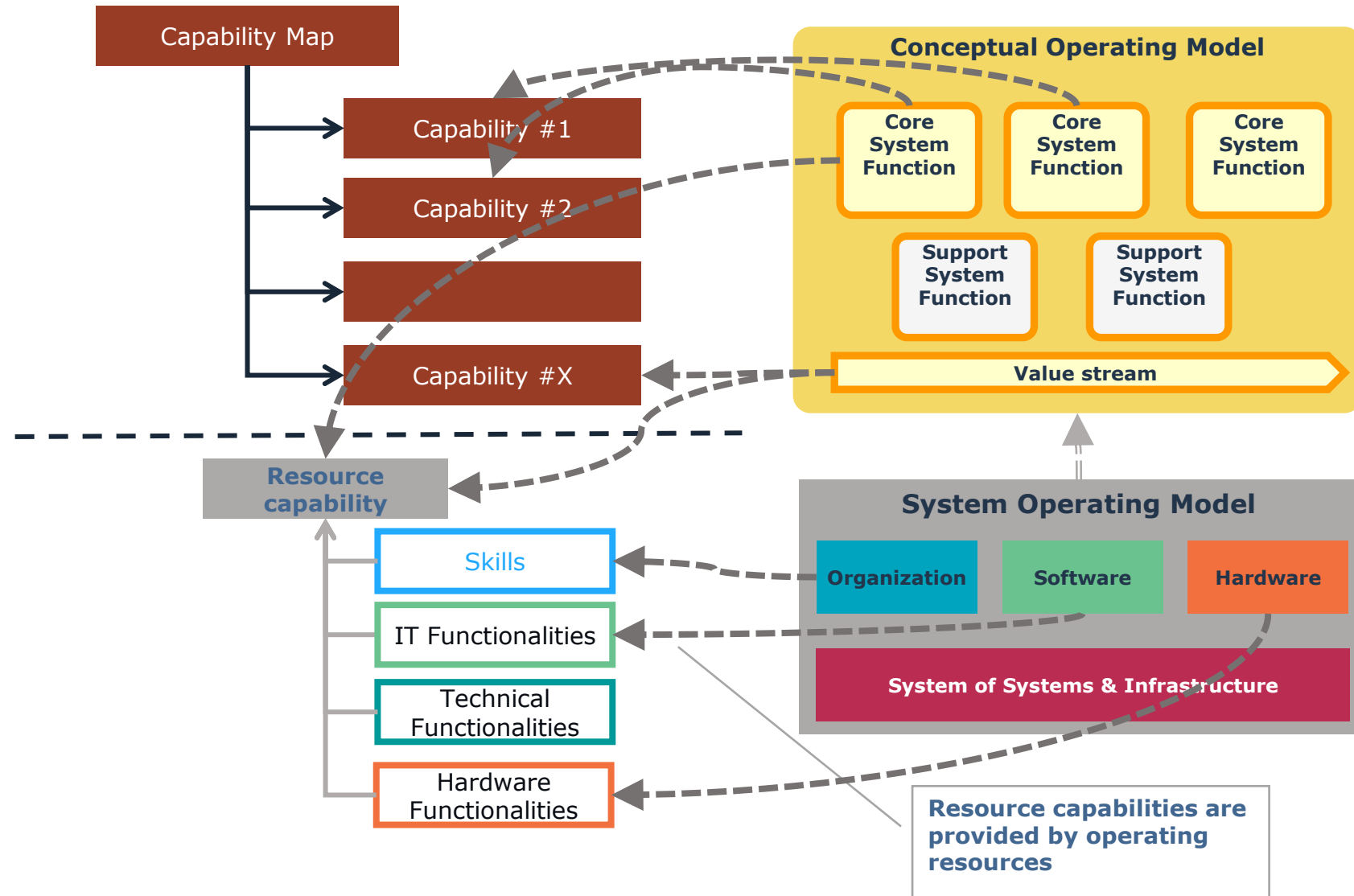
- Capabilities are fulfilled by System Functions.
- System Functions required Functionalities/Skills
- Value Streams require Capabilities



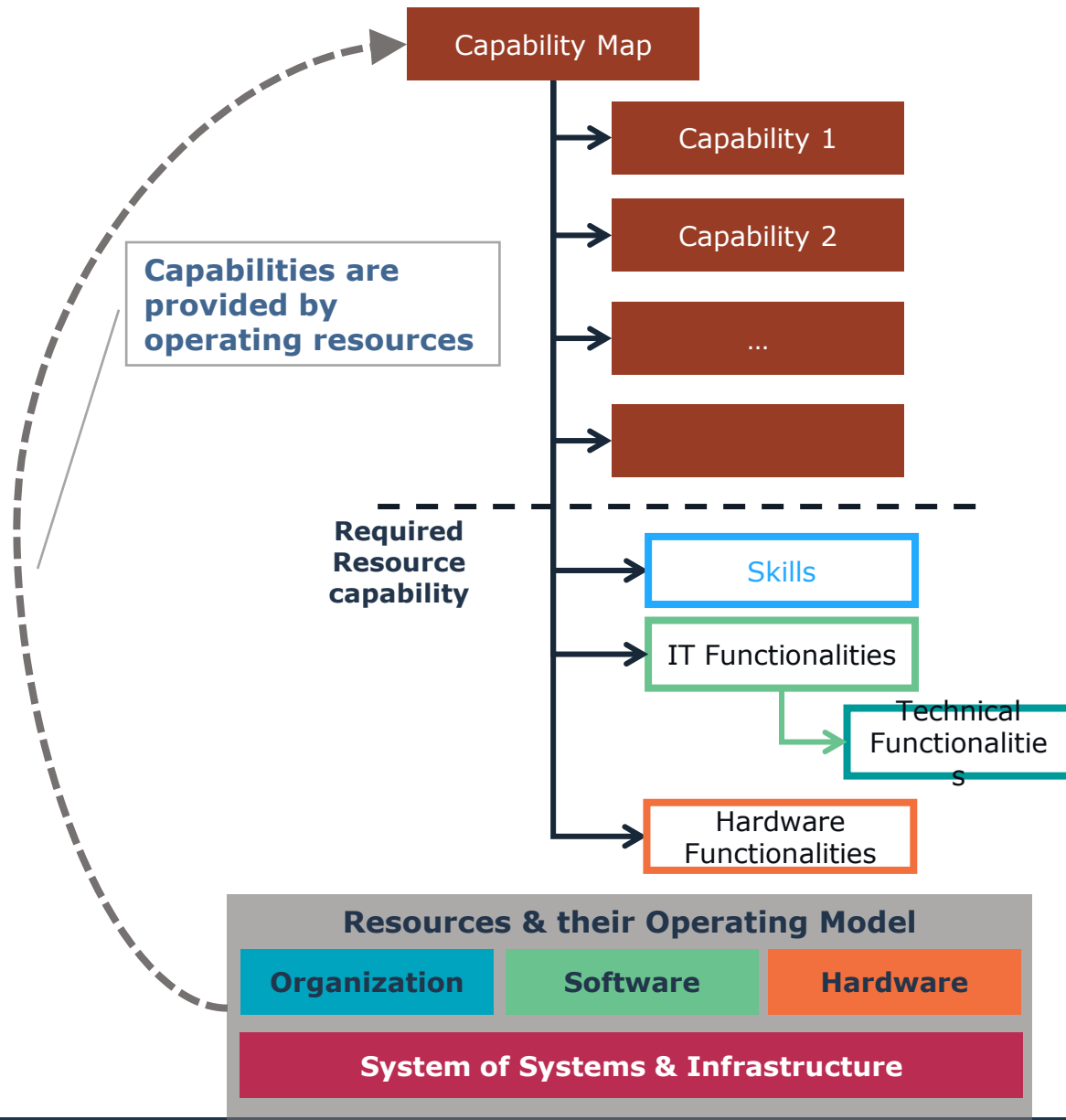
Strategic Alignment of Resources Capabilities

- Capabilities are fulfilled by System Functions.
- System Functions required Functionalities/Skills
- Value Streams require Capabilities
- System Functions and related functionalities are realized by operating Resources

→ Flexible,
modular,



Alternative Capability single breakdown



Capability decomposition goes down to functionalities and skills → No differentiation between enterprise strategic capabilities (Capability) and enterprise operating models (System Functions).

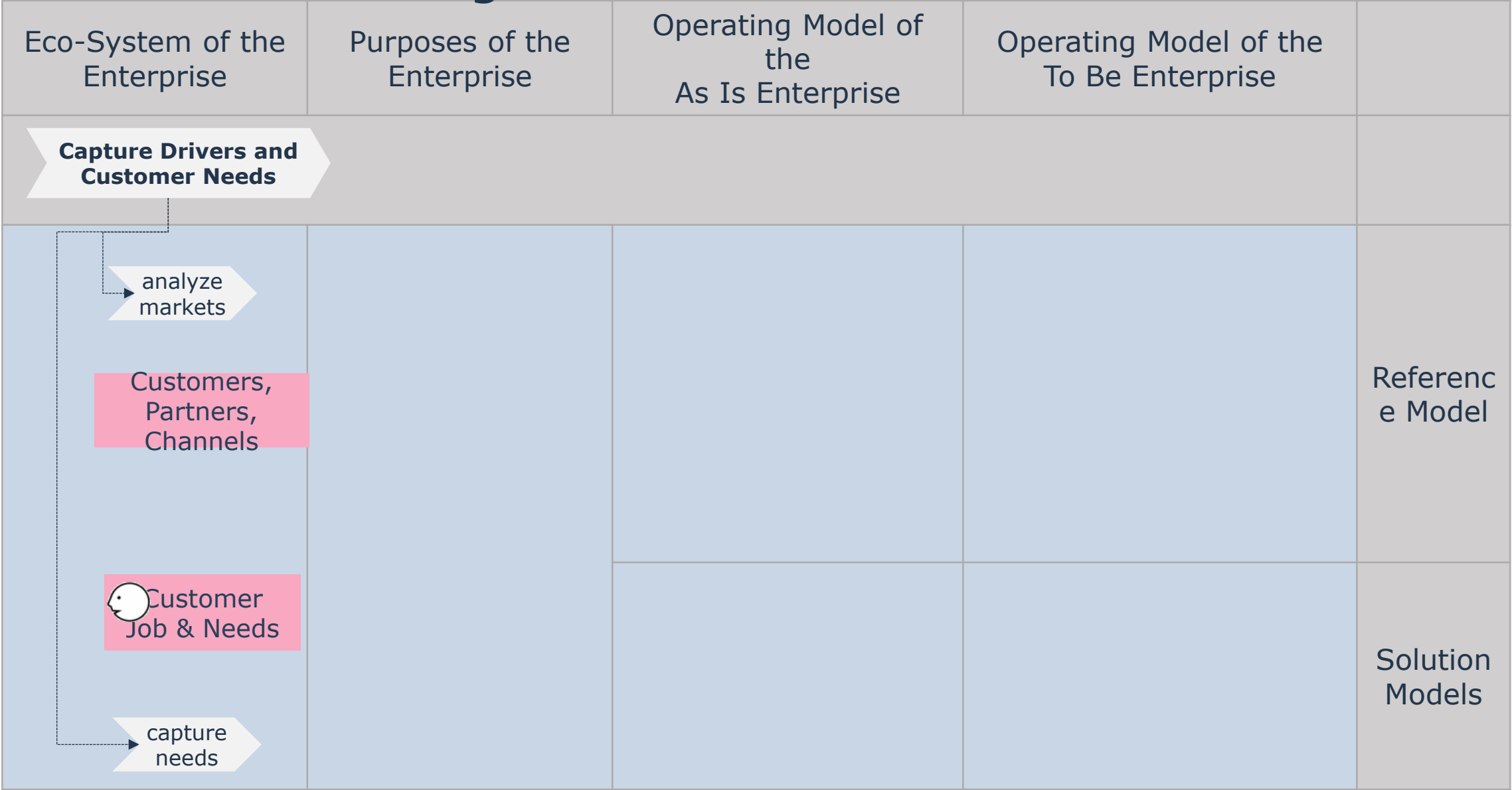
-> Preferential approach for US.

-> Major limitation: amalgamation of Capability and resource capability.

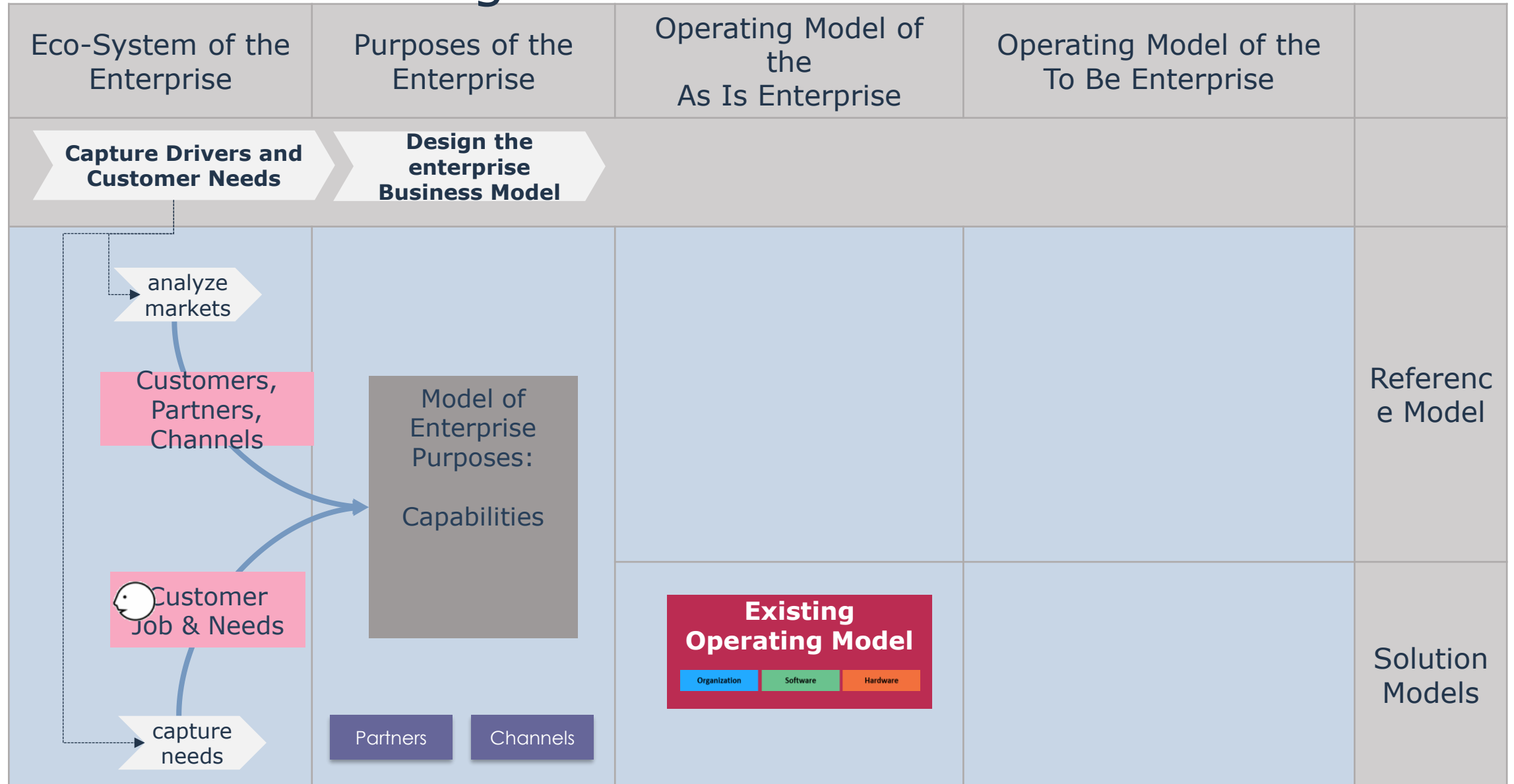
An improved approach is to distinguish Resource-Capabilities from Business-Capabilities.

- Simple but hierarchical & rigid
- Risk of confusing "Capabilities" and Resource Capabilities" with "Capability levels"..

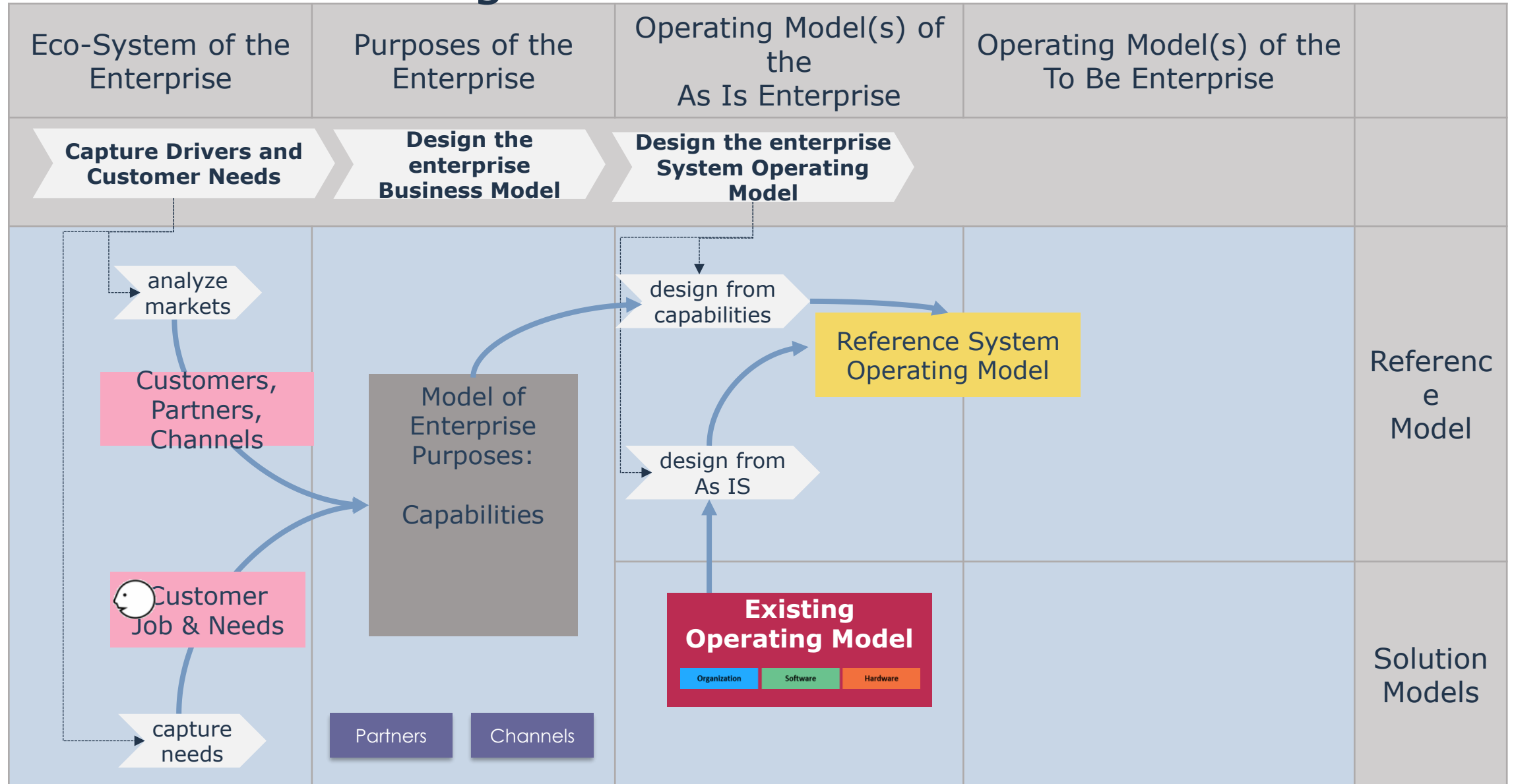
Transformation Design Method Overview



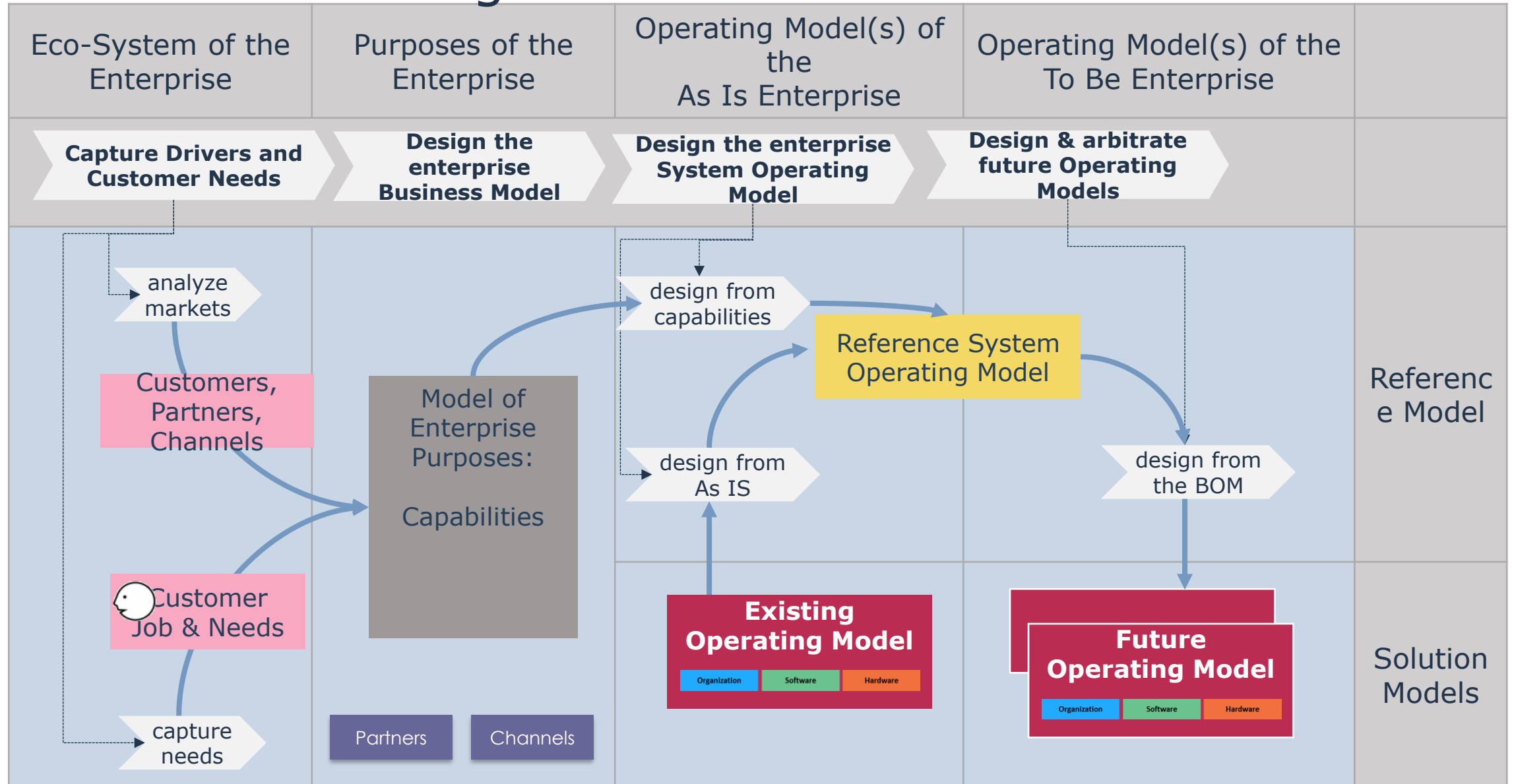
Transformation Design Method Overview



Transformation Design Method Overview

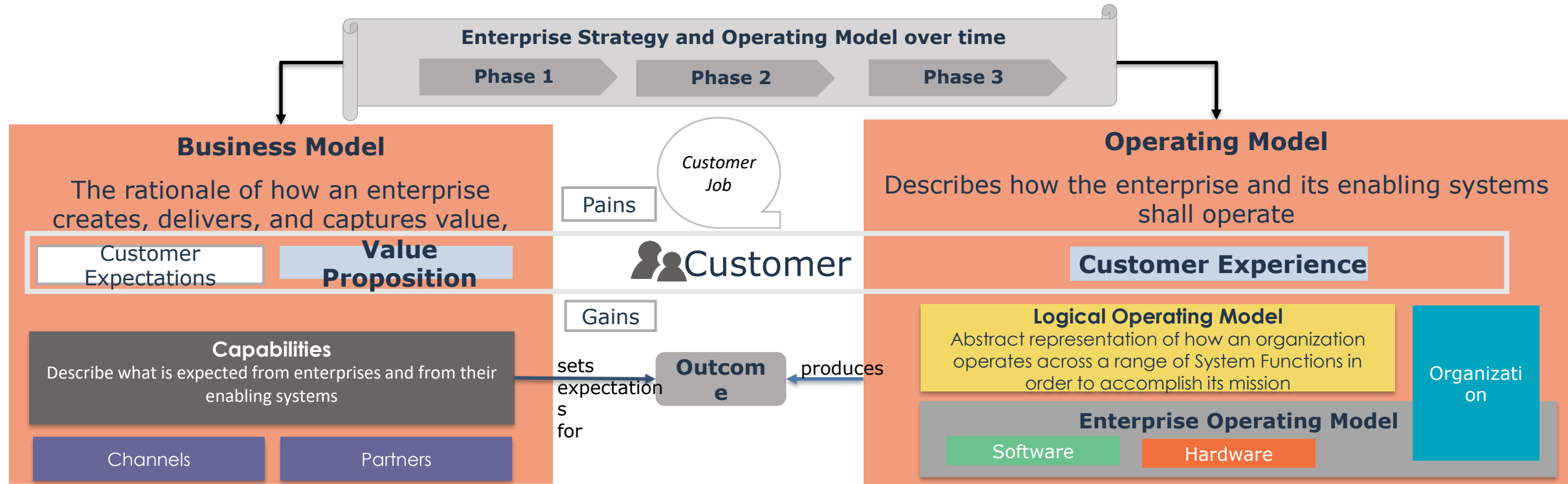


Transformation Design Method Overview



Business Model vs System Operating Model

- Enterprise Business Model
 - A business model describes the rationale of how an organization creates, delivers, and captures value, in economic, social, cultural or other contexts. The process of business model construction is part of business strategy.
 - In a Capability centric approach, the business model is the planning of Capabilities and their associated value propositions.
- Enterprise Operating Model
 - An operating model is an abstract representation of how an organization operates across a range of domains in order to accomplish its mission. There are two major kinds of operating models:
 - Conceptual business model often called "System Operating Model".
 - Physical operating models: Organizational operating model, IS operating model, Infrastructure operating model, ...

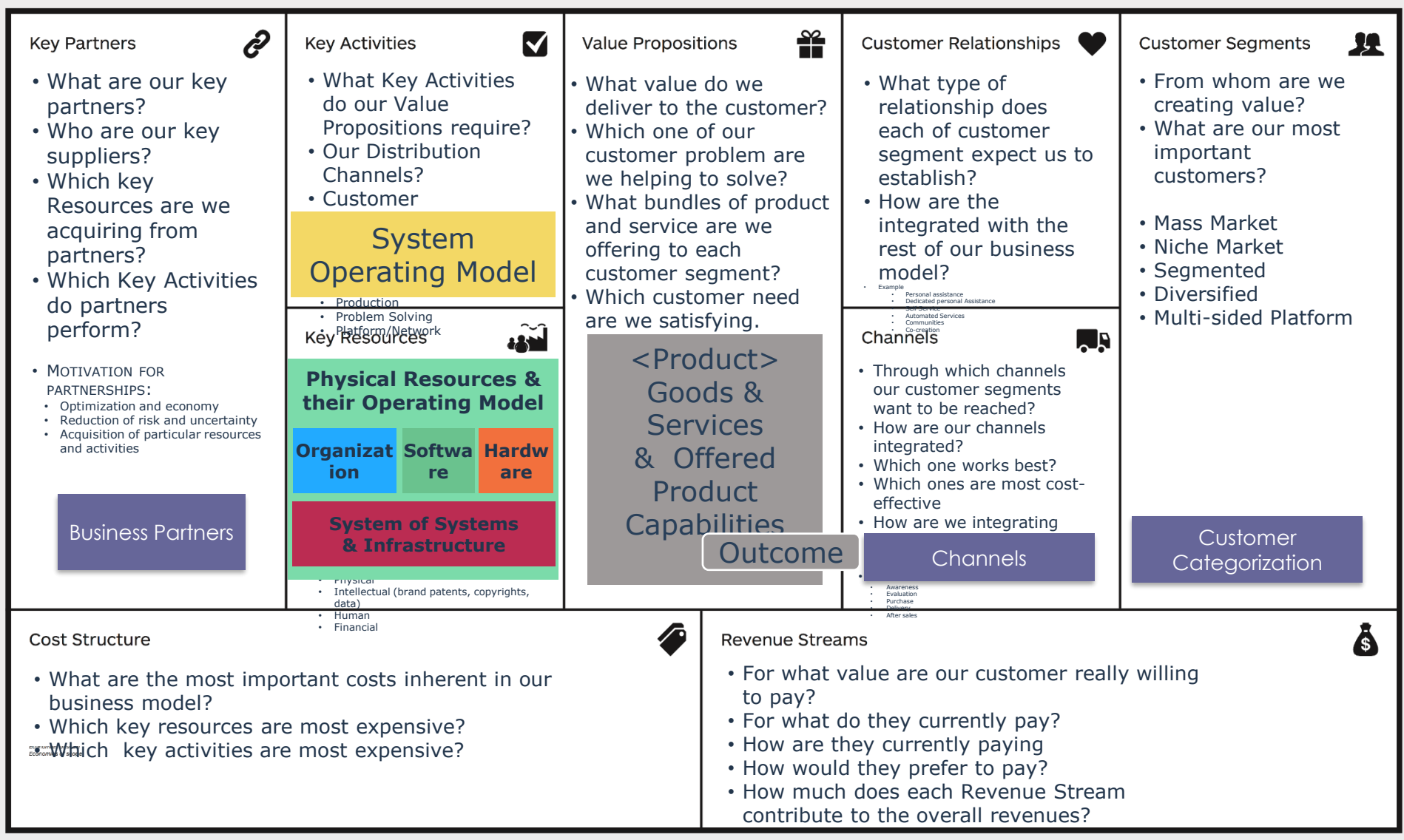


Mapping to management tools

Business Model Canvas

Value Proposition Canvas

Business Model Canvas Mapping



Business Model Canvas – Revenues & Costs Details

Cost Structure  • What are the most important costs inherent in our business model? • Which key resources are most expensive? • Which key activities are most expensive? • How much does each Revenue Stream contribute to the overall revenues?	Revenue Streams  • For what value are our customer really willing to pay? • For what do they currently pay? • How are they currently paying • How would they prefer to pay? • How much does each Revenue Stream contribute to the overall revenues?
---	--

IS YOUR BUSINESS MORE

- Cost driven (leanest cost structure, low price value proposition, maximum automation, extensive outsourcing)
- Value driven (focused on value creation, premium value proposition)

SAMPLE CHARACTERISTICS

- Fixed costs (salaries, rents, utilities)
- Variable costs
- Economies of scale
- Economies of scope

TYPES

- Asset Sale
- Usage fee
- Subscription fees
- Lending/Renting/Leasing
- Licensing
- Brokerage fees
- Advertising

FIXED PRINCING

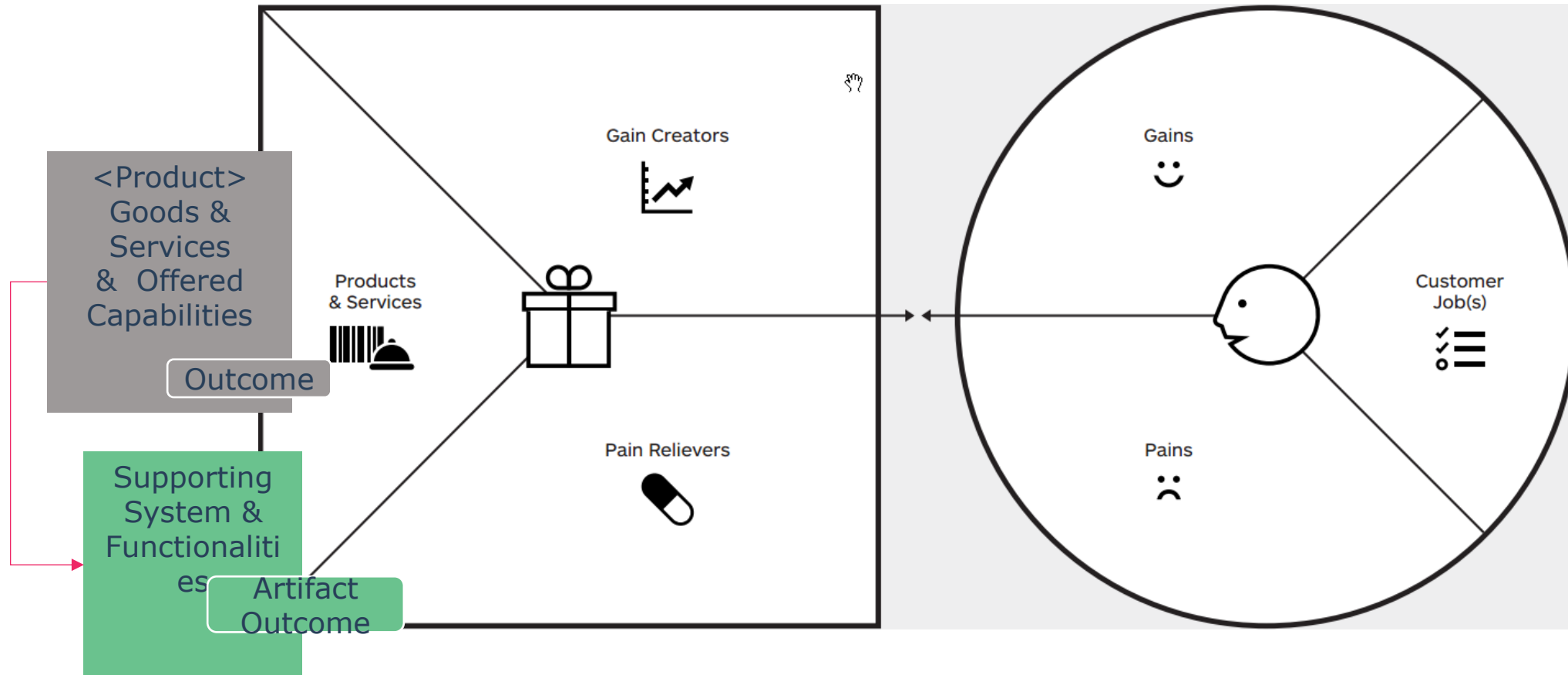
- List price
- Product feature dependent
- Customer segment dependent
- Volume dependent

DYNAMIC PRICING

- Negotiation (bargaining)
- Yield management
- Real-time market

Value Proposition Canvas Mapping

Each feature includes a benefit hypothesis and acceptance criteria,



[Job to be done is included in:
HOPEX Metamodel Overview \(demomega.com\)](https://demomega.com)