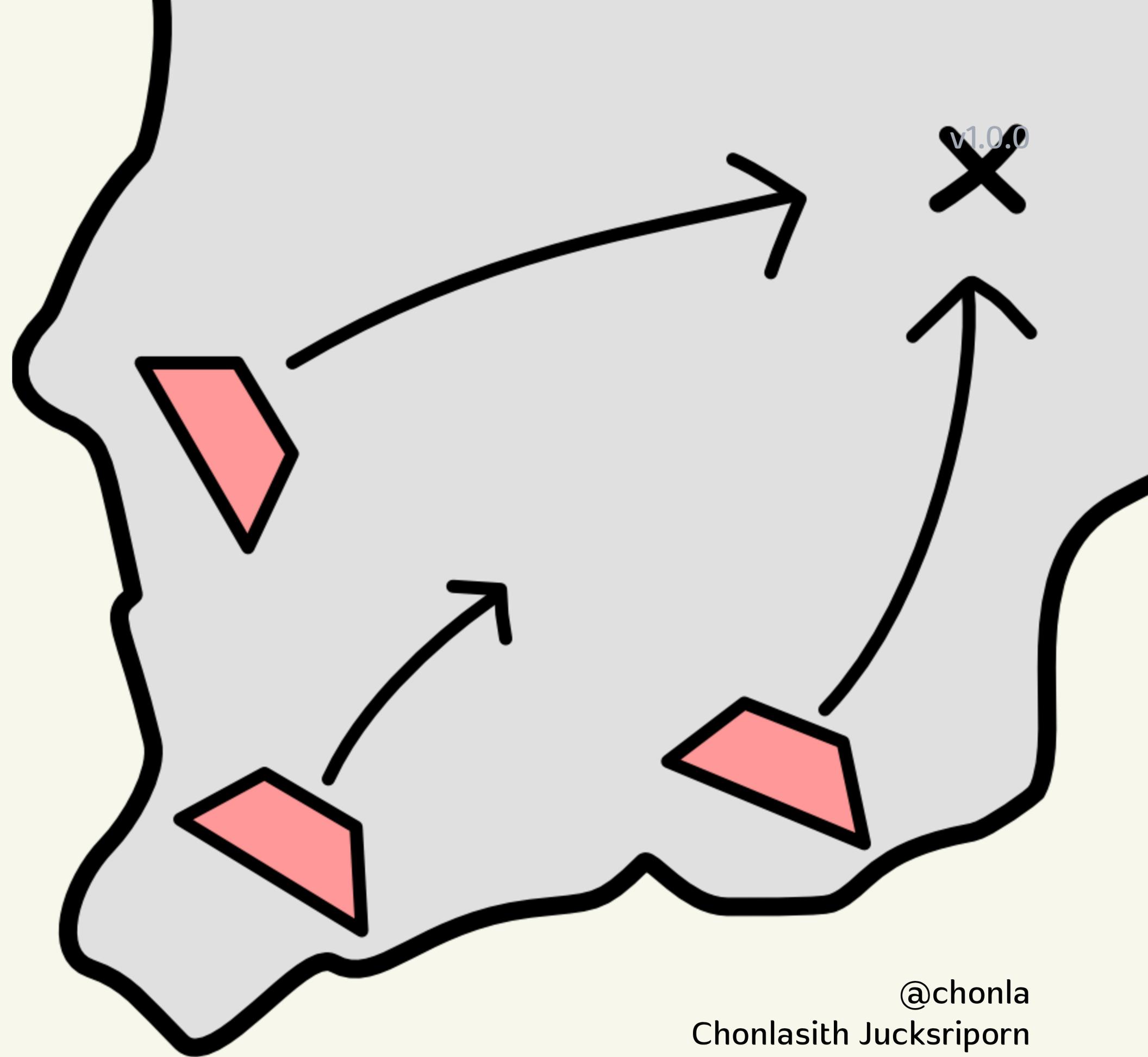




Wardley Map for X

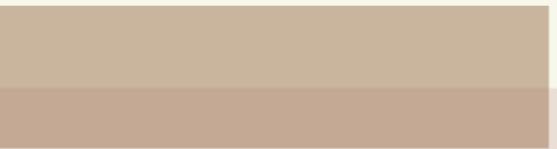


12 February 2025

@chonla
Chonlasith Jucksriporn

กลุ่มการเรียนรู้

กลุ่มละ 4-5 คน



ชื่อกลุ่ม + ชื่อสมาชิก



จัดกลุ่ม



ความคาดหวัง

- **สร้าง** Wardley Map ได้เอง
- **อธิบาย** ถึง Why for Purpose และ Why for Movement ได้
- **อธิบาย** ถึงการใช้ Wardley Map ร่วมกับ Climatic Patterns ได้
- **อธิบาย** ถึงการใช้ Wardley Map ร่วมกับ Doctrines ได้

ເນື້ອຫາ

- ປະລຸມບທ
- Landscape
- Climate
- Doctrine
- Gameplay
- OODA
- Anti patterns

3 คำダメ

- ปัจจัยอะไรที่เราใช้ในการตัดสินใจได ๆ
- ประโยชน์ของแผนที่คืออะไร
- ทำไมเราต้องมีแผนที่



เขี๊ยะขอสรุป
ของกลุ่ม



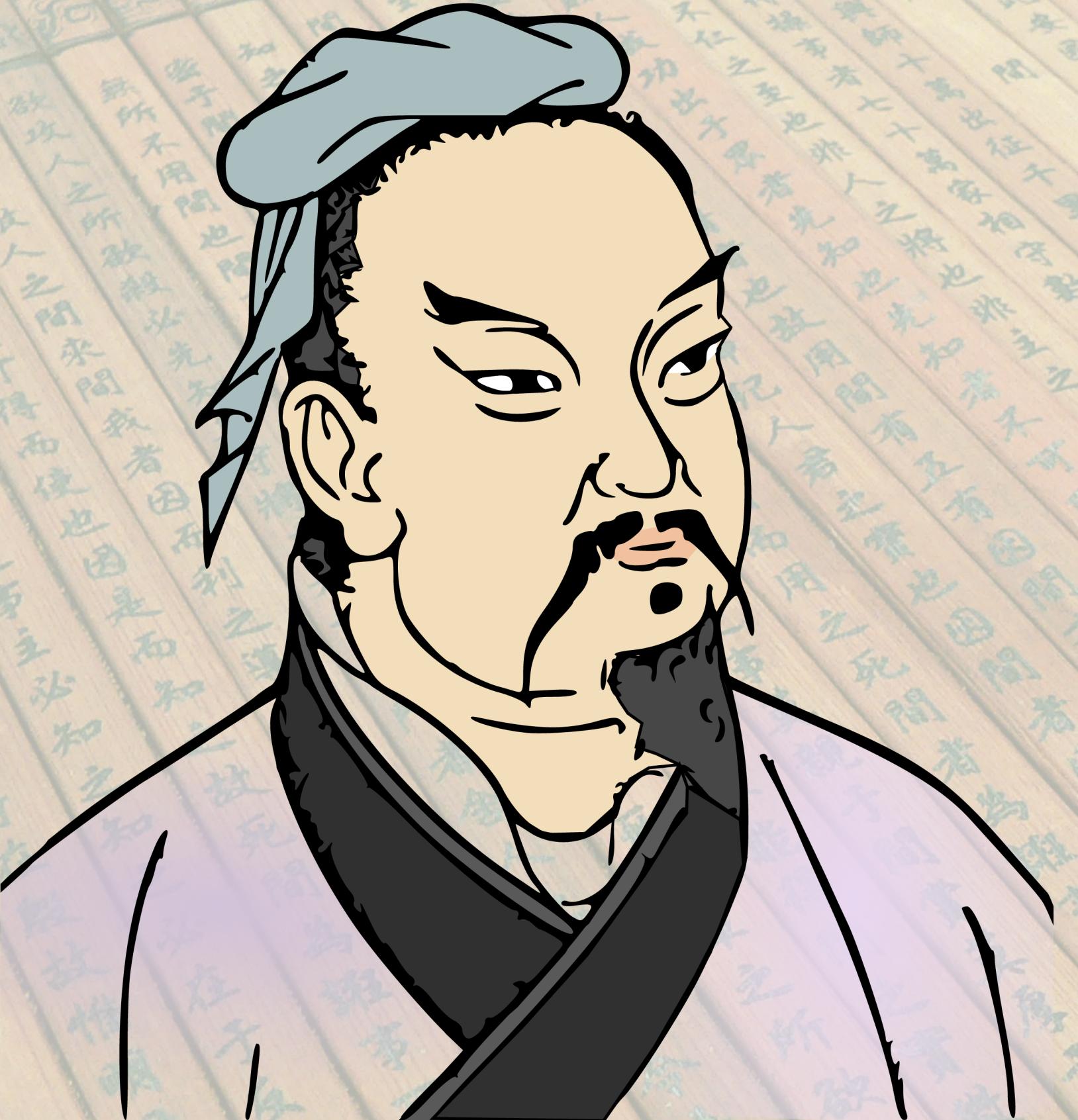
แบ่งปันข้อสรุป

- ปัจจัยอะไรที่เราใช้ในการตัดสินใจได้ ๆ
- ประโยชน์ของแผนที่คืออะไร
- ทำไมเราต้องมีแผนที่



แบ่งปันข้อสรุป
ของกลุ่ม



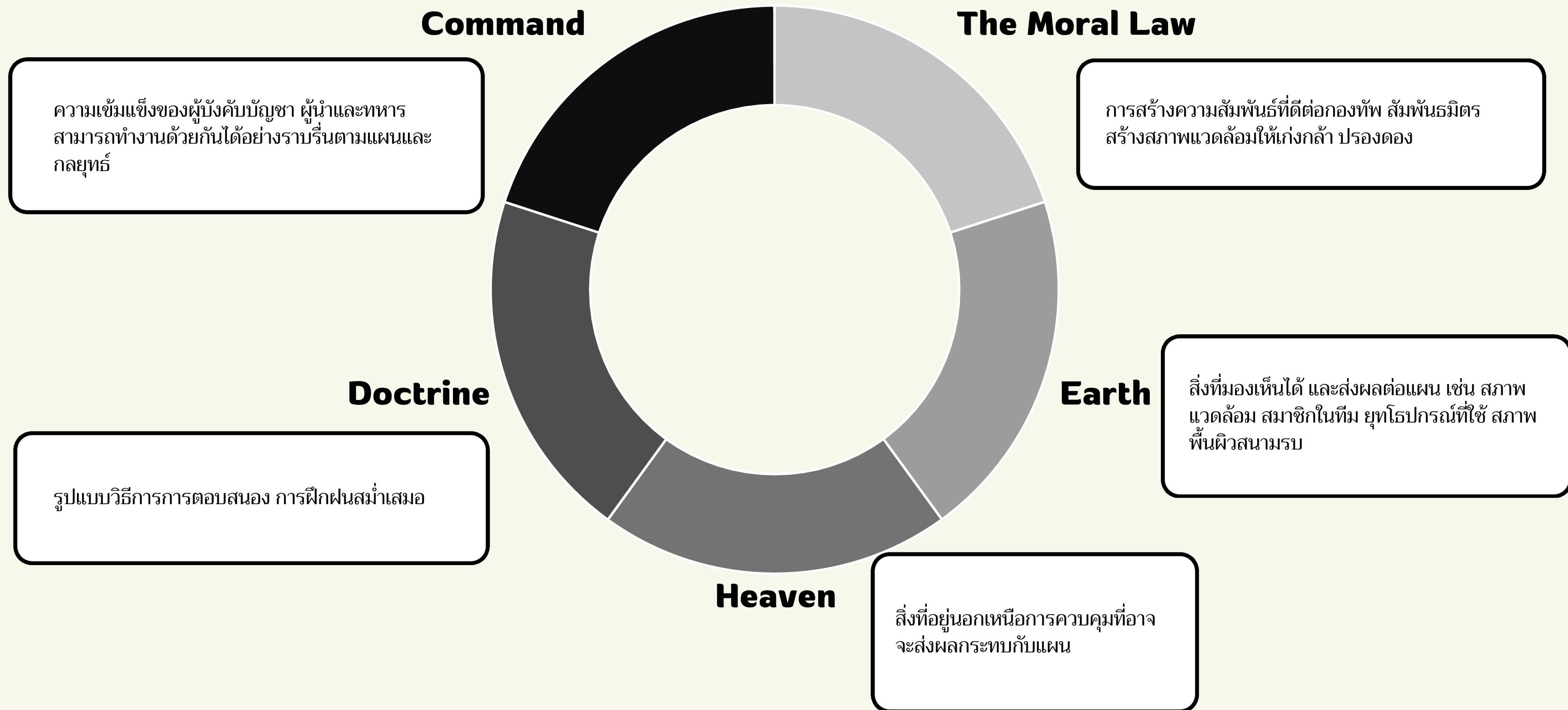


孫子兵法

THE ART OF WAR - SUN TZU

ตำราพิชัยสงครามชุนวุ

ปัจจัย 5 ประการที่จะทำให้มีชัยในการศึก



ນີ້ຈະໄຮຍ່ໃນປັຈວັຍໃຫນບ້າງ ດ້ວຍເຫດຸພລວະໄຣ

- ປັຈຈໍຍອະໄຣທີ່ເຮົາໃຊ້ໃນການຕັດສິນໃຈໄດ້ ↙
- ປະໂຍ່ຈນໍຂອງແຜນທີ່ຄືອອະໄຣ
- ທຳມະເຮົາຕ້ອງມີແຜນທີ່



ໜ້າ
ຈັດກາລຸມ



แบ่งปันข้อสรุป

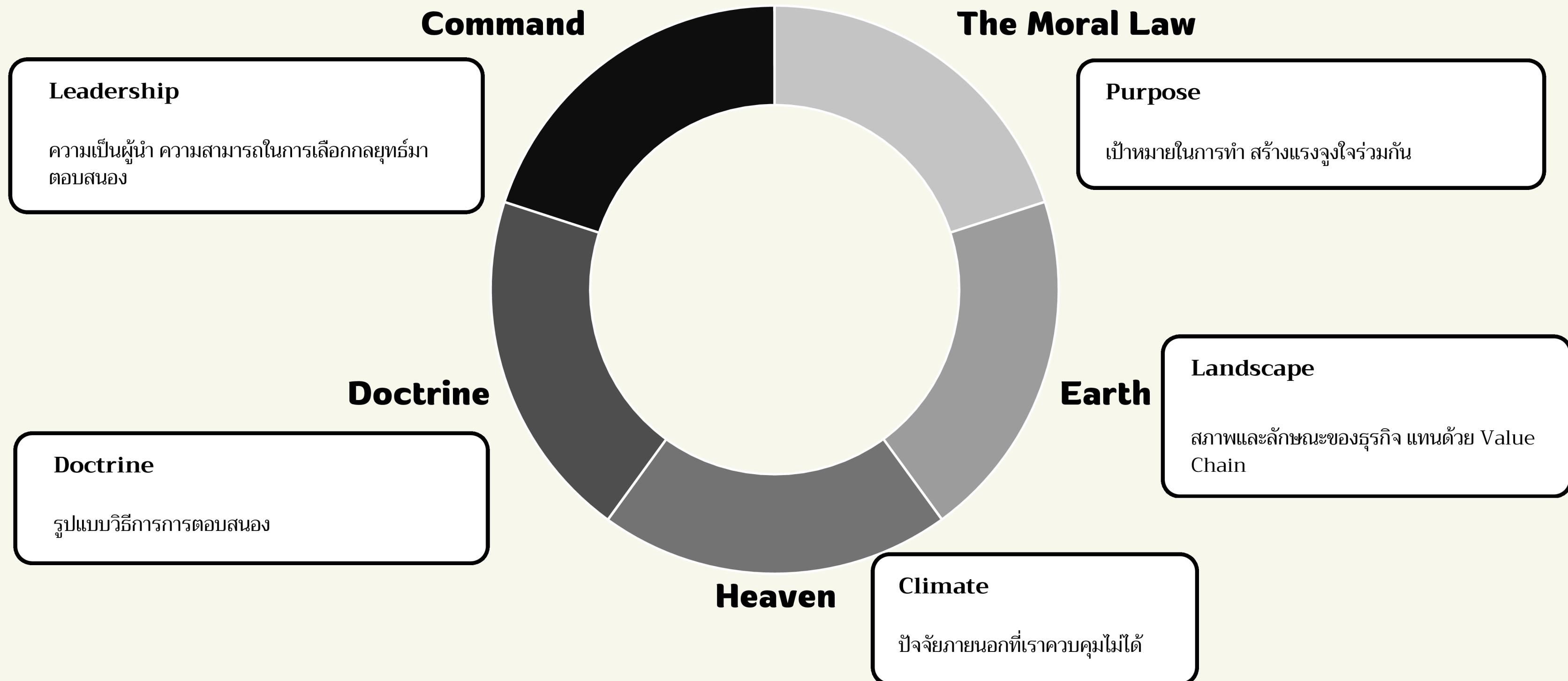
- ปัจจัยอะไรที่เราใช้ในการตัดสินใจได้ ๆ
- ประโยชน์ของแผนที่คืออะไร
- ทำไมเราต้องมีแผนที่



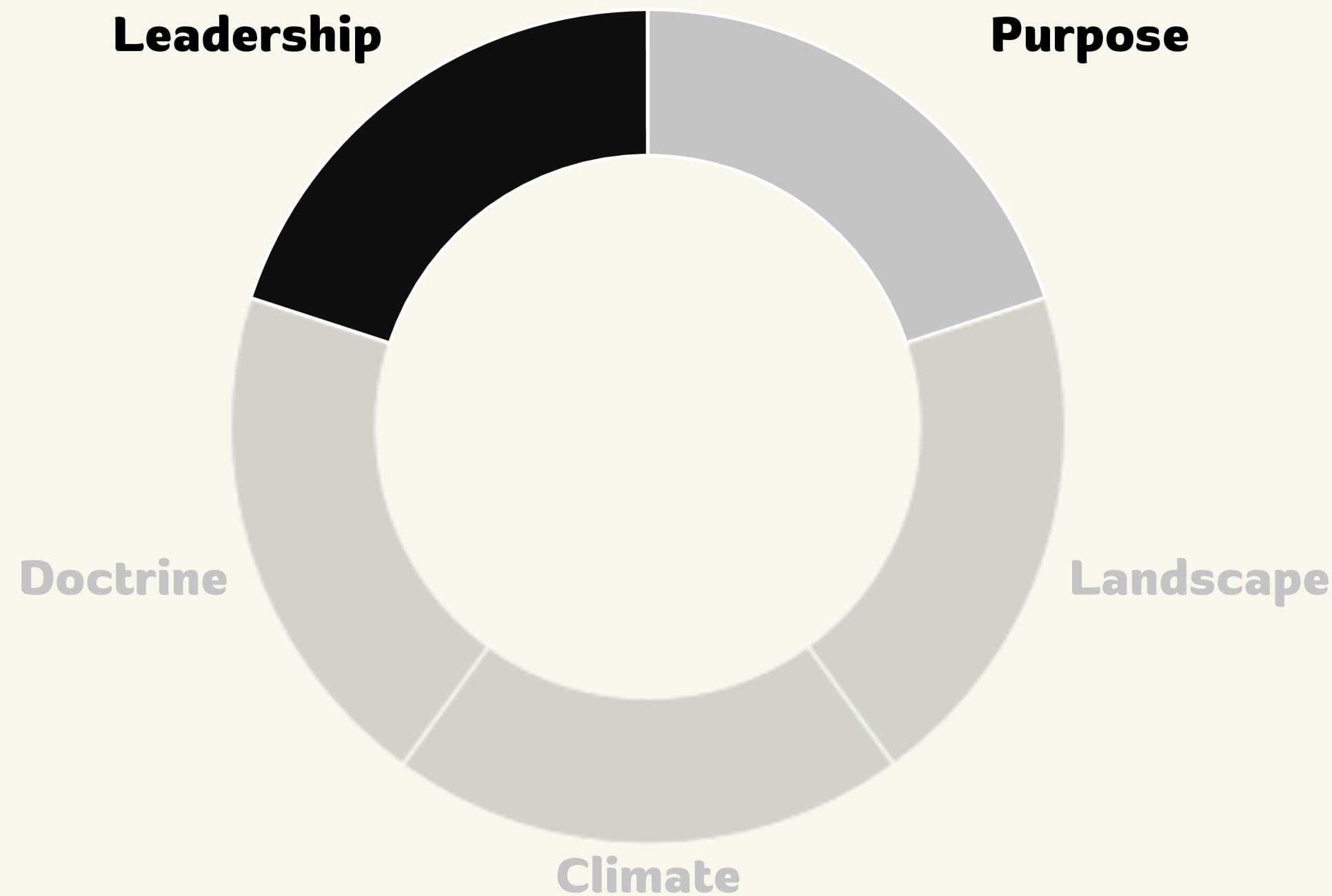
แบ่งปันข้อสรุป
ของกลุ่ม



ปัจจัย 5 ประการที่จะทำให้มีชัยในการศึก



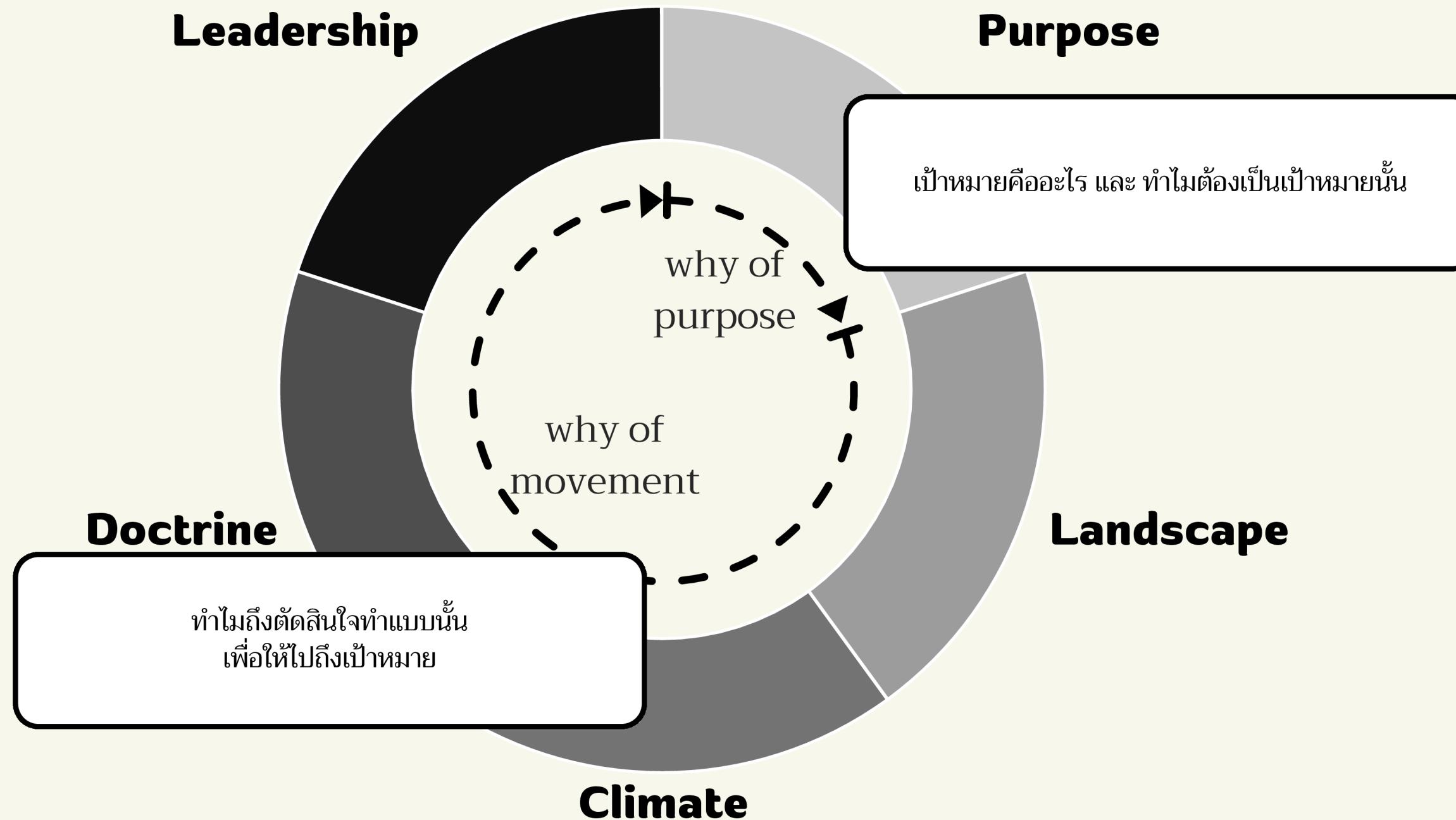
Strategy by gut feel



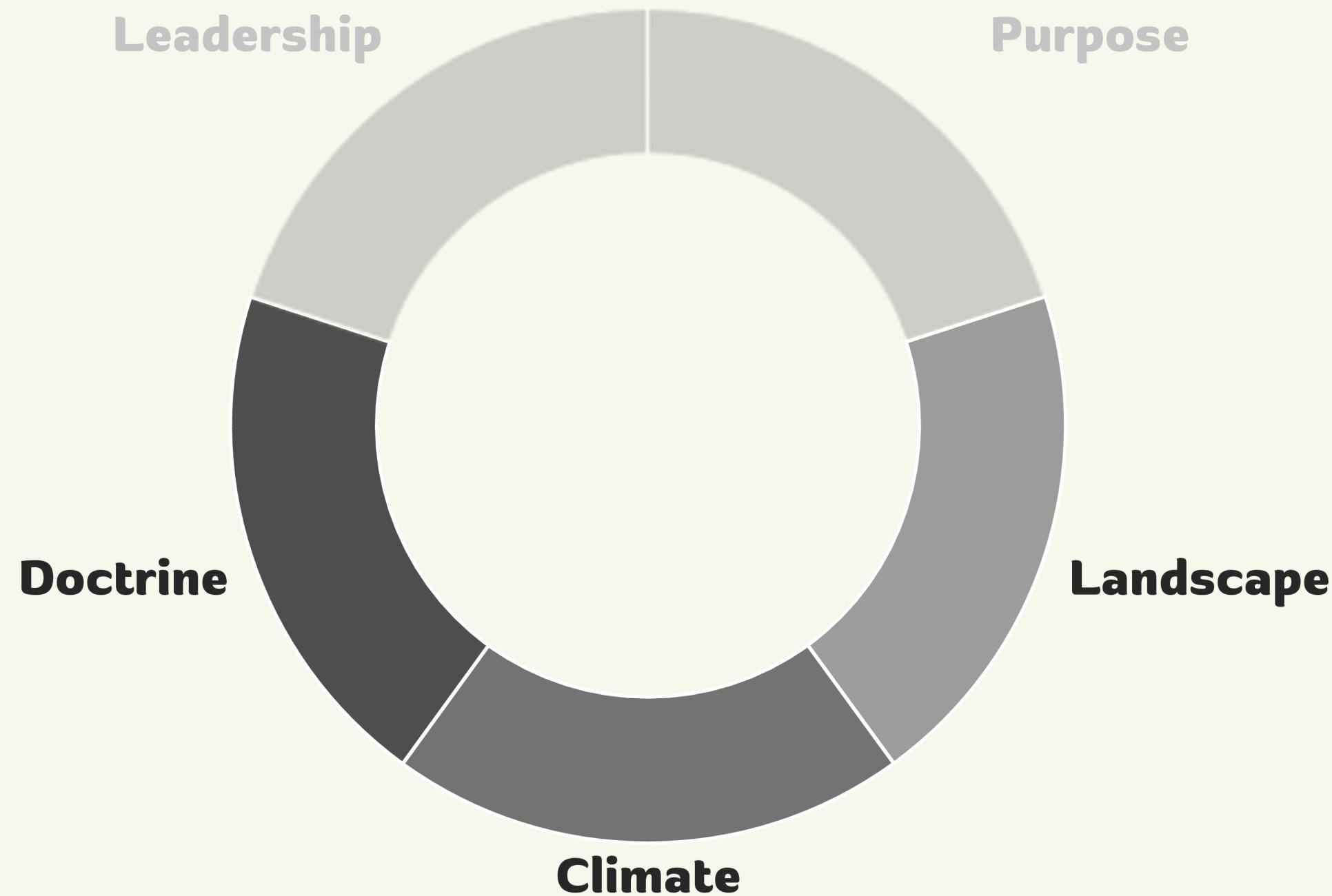
Strategy by gut feel



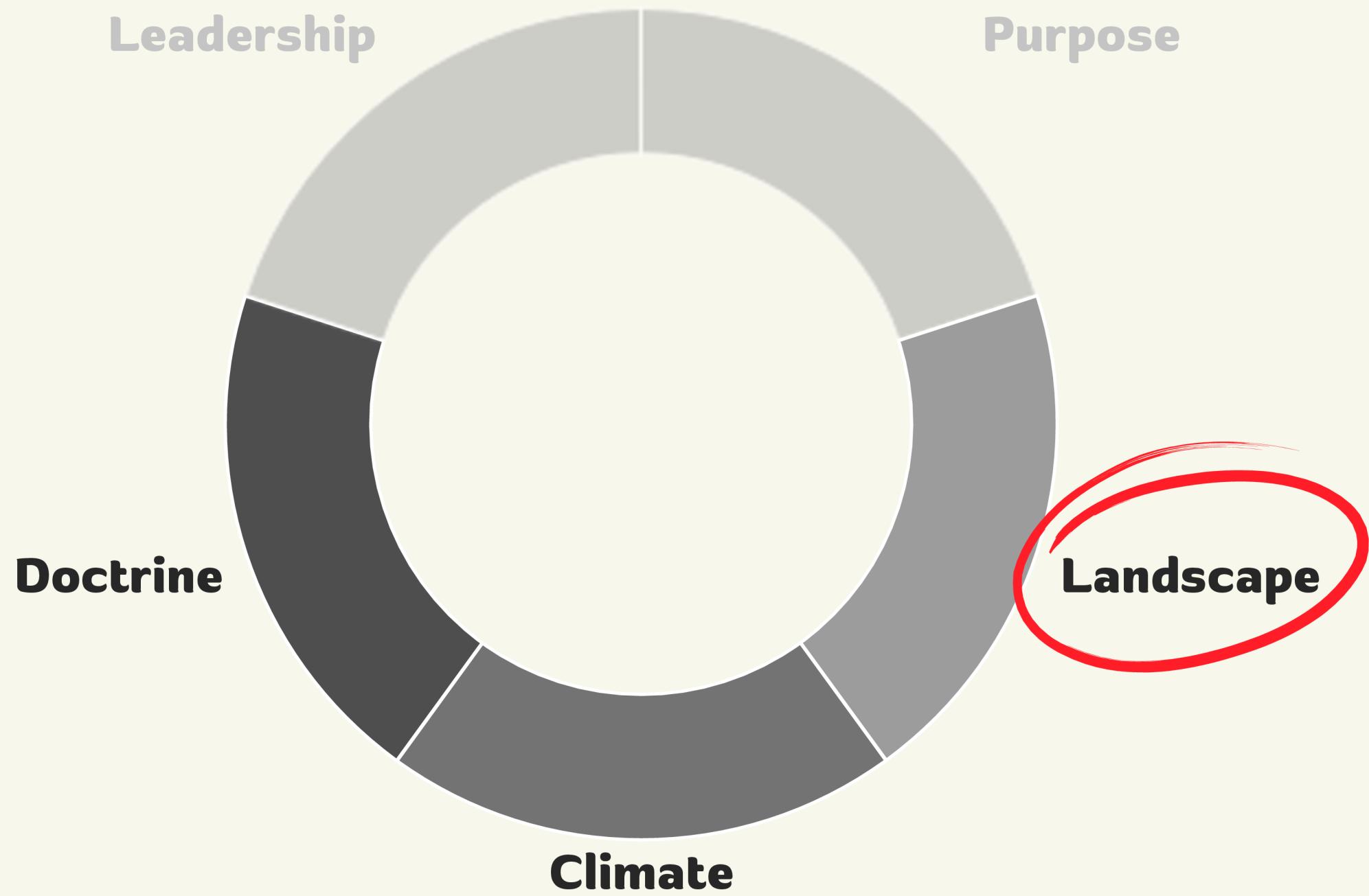
The 2 WHYs



The 3 missing pieces



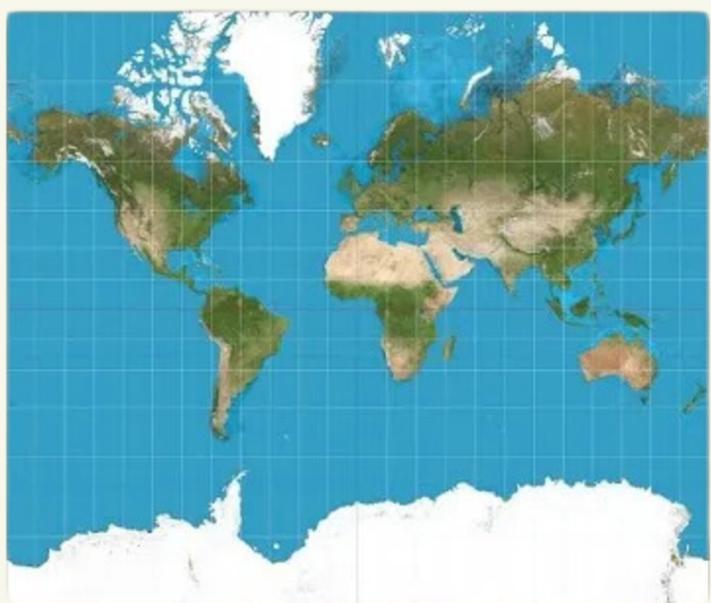
Landscape



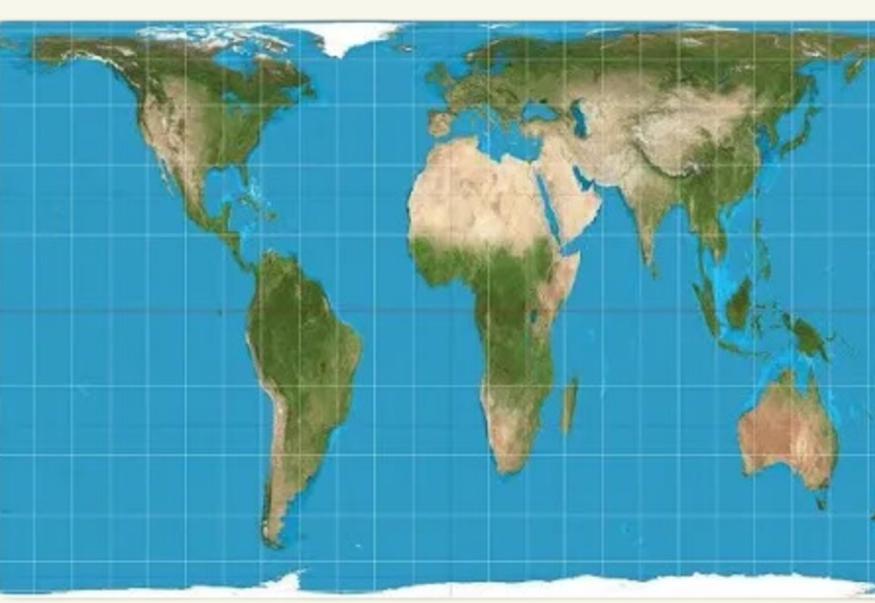
Landscape ແນ້ນດ້ວຍ Map

ແຜນທີ່ (Map)

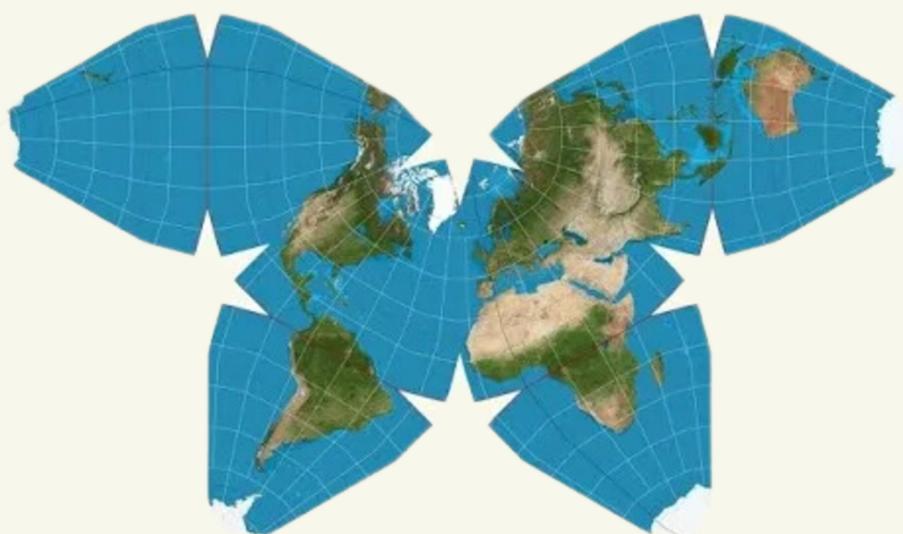
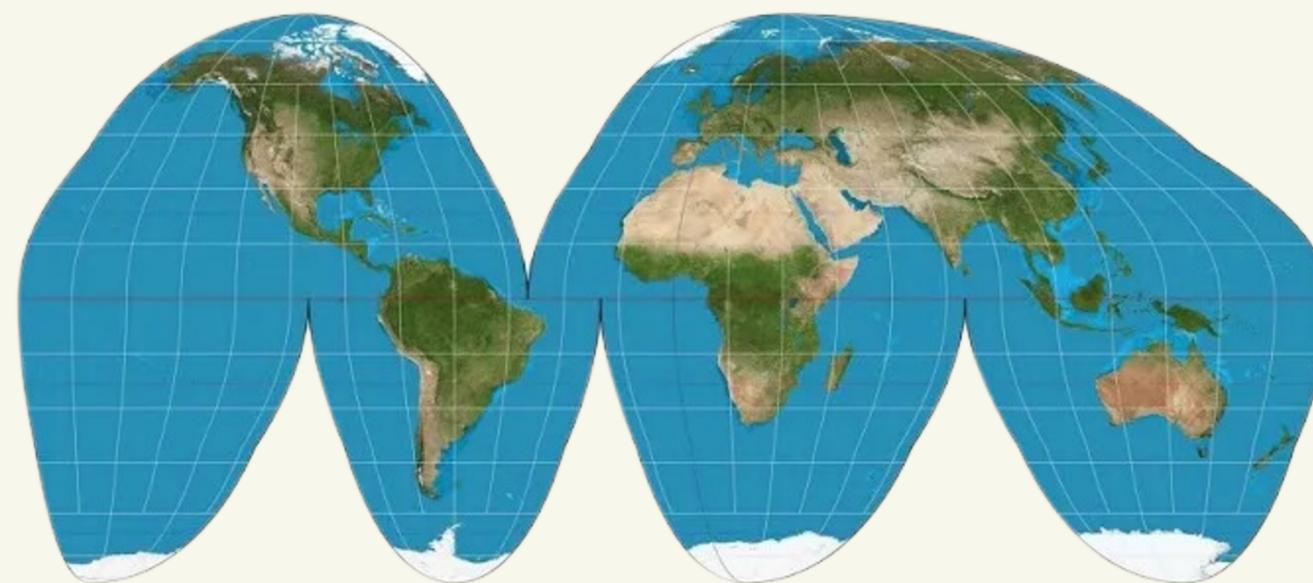
MERCATOR



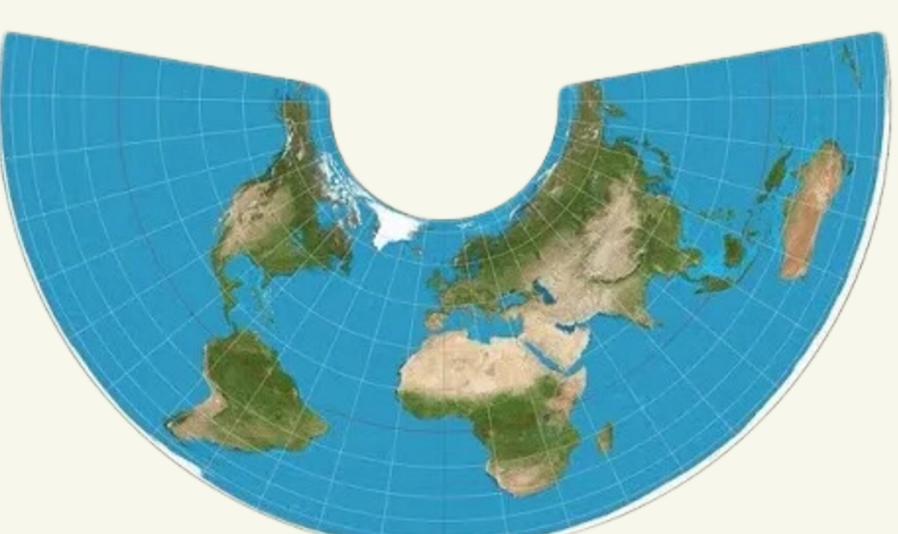
GALL-PETERS



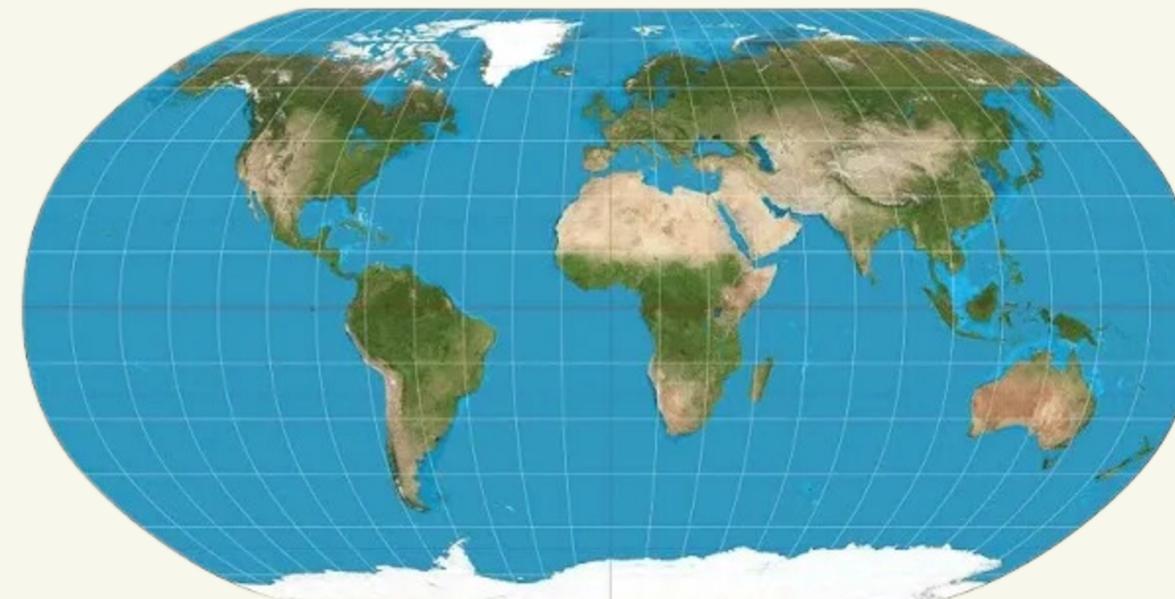
GOODE-HOMOLOSIONE



WATERMELON



ALBERS



ROBINSON

แผนที่บอกระไรเรา

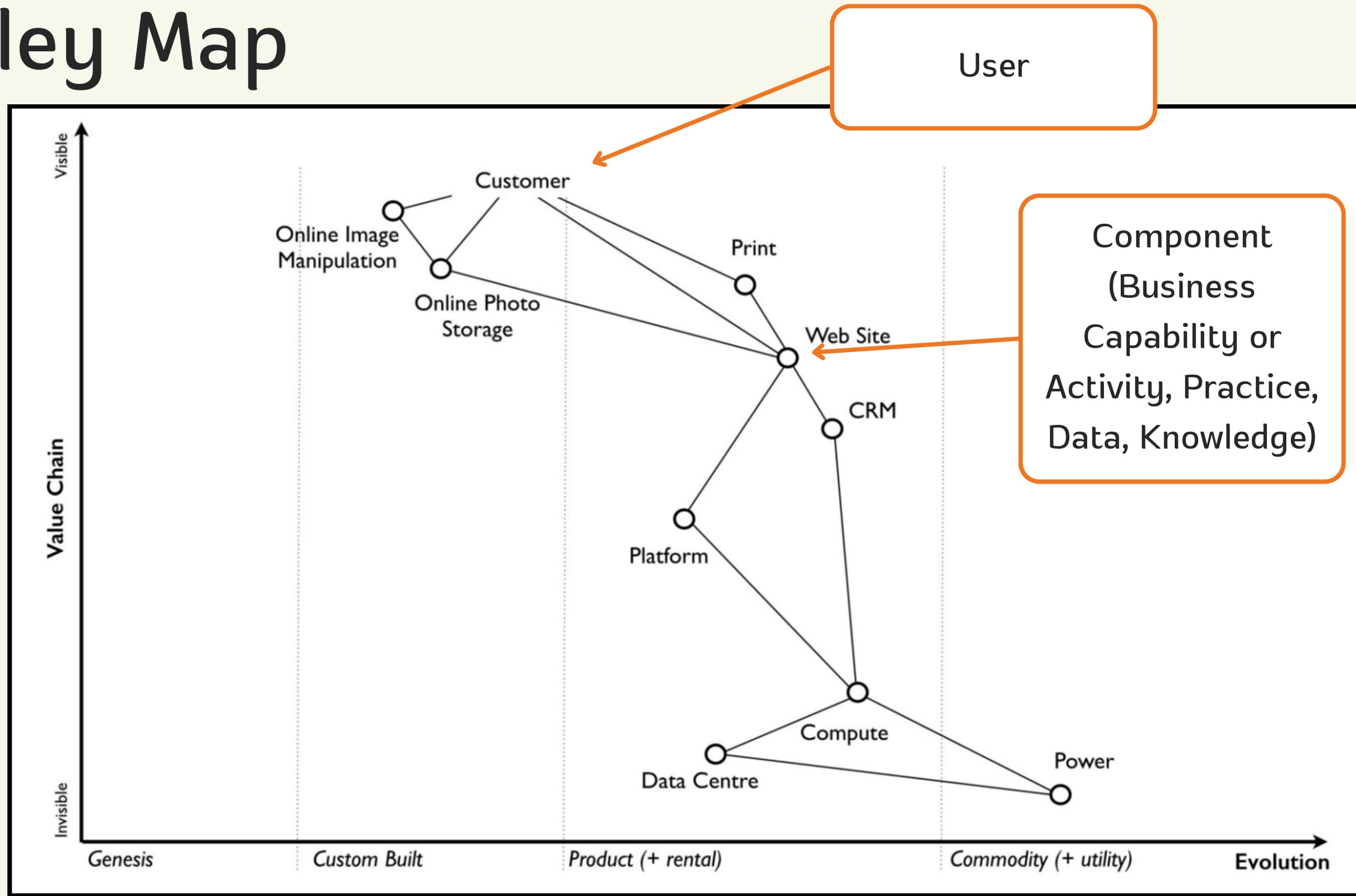
แผนที่ จะมีประโยชน์ตอนที่เราเอามันมาใช้

- **Visual Representation** Visual สื่อสารได้ดีกว่า Textual หรือ Verbal มาก
- **Context Specific** ให้เราเห็นสถานการณ์ในเหตุการณ์ที่เฉพาะลงไป
- **Components** ให้เราเห็นส่วนประกอบต่าง ๆ ที่เกี่ยวข้องของ Context
- **Anchor** ให้เราเห็นเป้าหมายที่เราจะไป
- **Movement** ให้เราเห็นทิศทางการเคลื่อนที่ของ Component
- **Position** ให้เราเห็นตำแหน่งของ Component ในแผนที่ในรูปแบบของความสัมพันธ์ของ Anchor และ Movement

Wardley Map

- แสดง **business landscape** เพื่อใช้ **share understanding**
- Anchor คือ **user** และ **need** ของ user
- กองทัพลูกแทนด้วย value chain ที่เป็น **capabilities** ที่จะตอบสนอง **need**
- Capabilities จะเรียงลำดับตาม **user visibility** และความสัมพันธ์ของ **capabilities** ด้วยกัน
- แกน Y จะแสดงถึง **user visibility** ยิ่งอยู่ใกล้ user แสดงว่า user จะเห็น
- แกน X จะแสดงถึง **stage ของ evolution**

Wardley Map



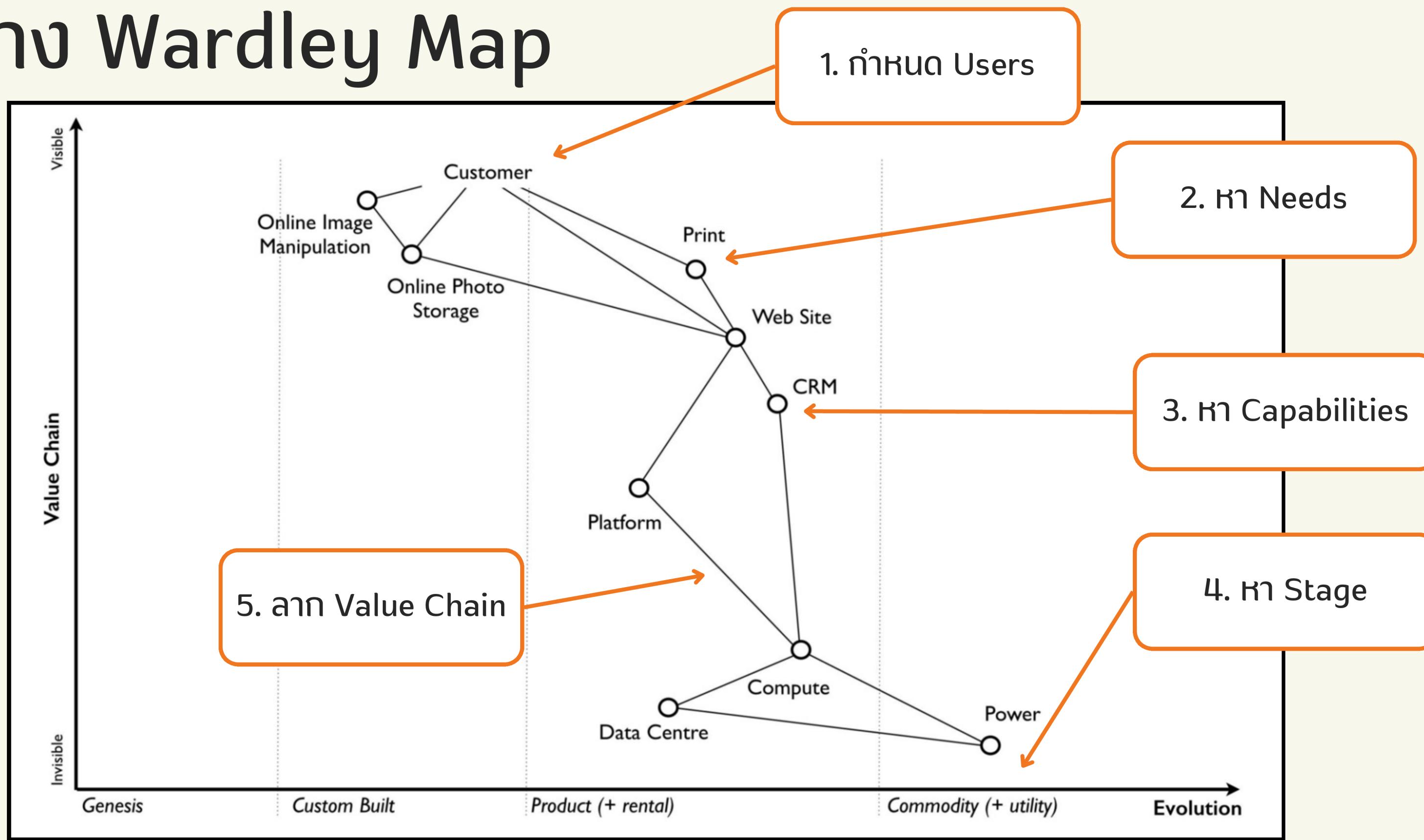
ឧបាណន៍តែង stages នៃវវ evolution

Type Stage of Evolution	I	II	III	IV
Activities	Genesis	Custom	Product + Rental Services	Commodity + Utility Services
Practices	Novel	Emerging	Good	Best
Data	Unmodelled	Divergent	Convergent	Modelled
Knowledge	Concept	Hypothesis	Theory	Accepted

วิธีสร้าง Wardley Map

1. กำหนด **User** ที่ต้องการ
2. หา **Needs** ที่ user ต้องการ เรียกว่า Top-level capabilities
3. หา **Capabilities** ทั้งหมดที่จะทำให้ Needs นั้นเกิดขึ้นได้จริง
4. ดูว่า Capabilities แต่ละอัน อยู่ใน **Stage ในของ Evolution** ถ้าคิดไม่ออก เพราะ Capabilities มันใหญ่ไป ลองแตกให้เล็กลง
5. ย้าย Capabilities ไปอยู่ใน **Evolutionary Stage** ที่คิดไว้ แล้วลาก **Value chain** ต่อ กันให้ครบ

វិធីសរ៍ាគ Wardley Map



Wardley Map ของฉัน

สร้าง Wardley Map ของงานที่ตัวเองกำลังทำ



Wardley Map ของฉัน

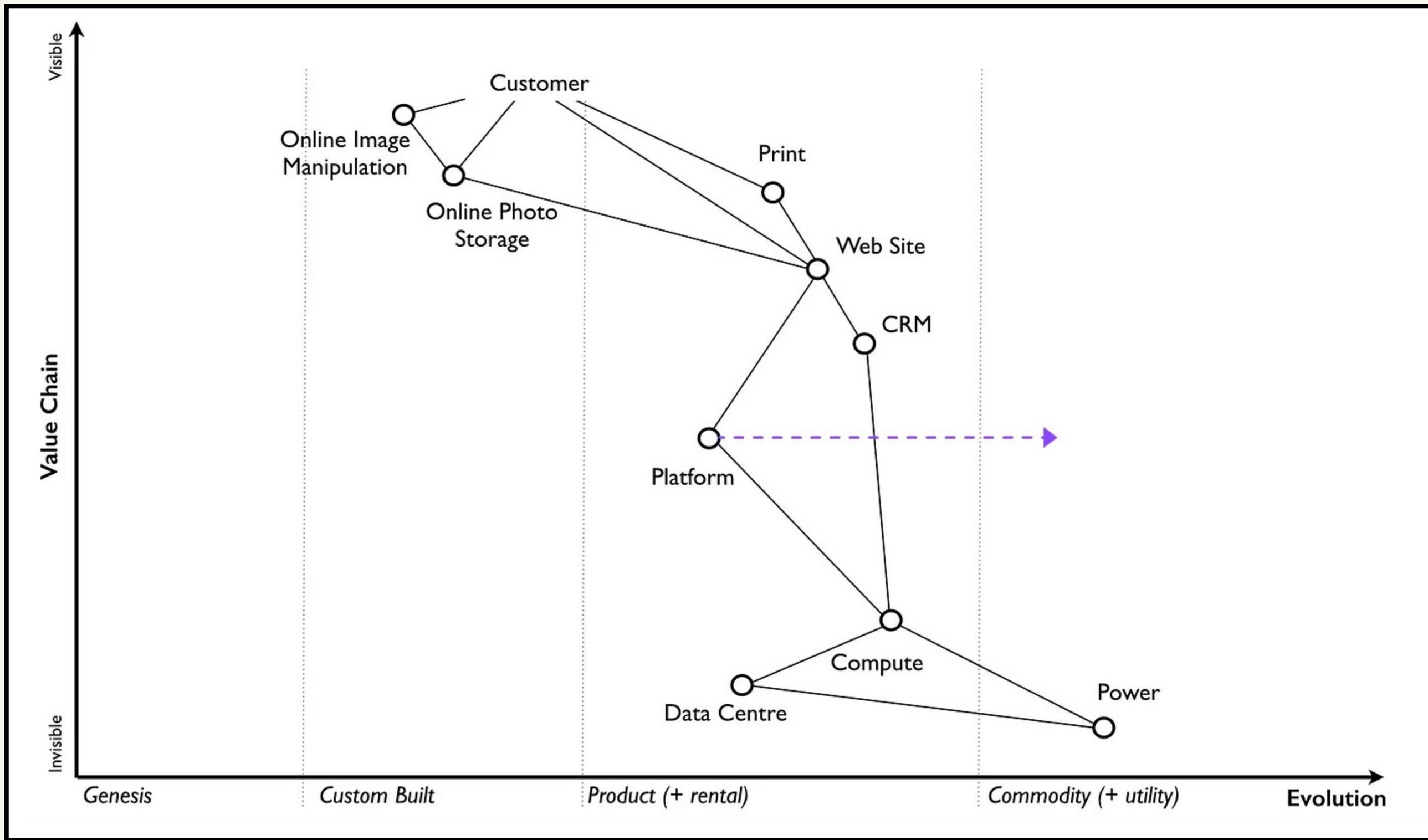
แม่งปั้น Wardley Map ของงานที่ตัวเองกำลังทำ



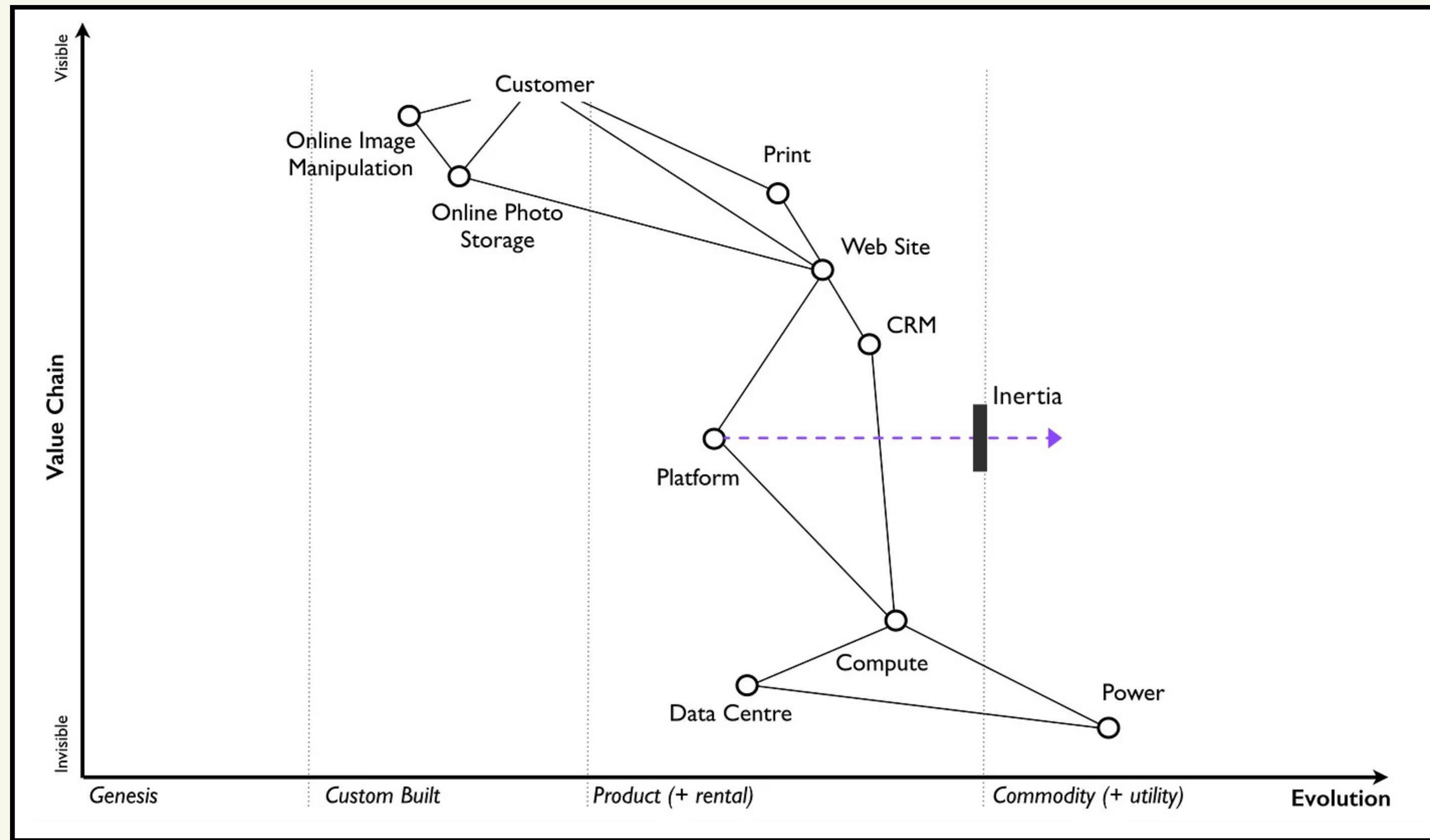
แม่งปັ່ນ
ໃຫກລຸມ



Evolution



Inertia



รับมือกับ Inertia

เราจะรับมือกับ Inertia อย่างไรได้บ้าง



Brainstorm

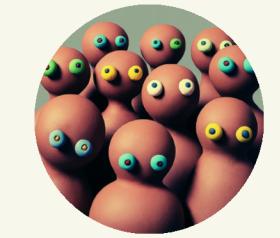


รับมือกับ Inertia

แบ่งปันวิธีการรับมือกับ Inertia



แบ่งปัน
ของกลุ่ม



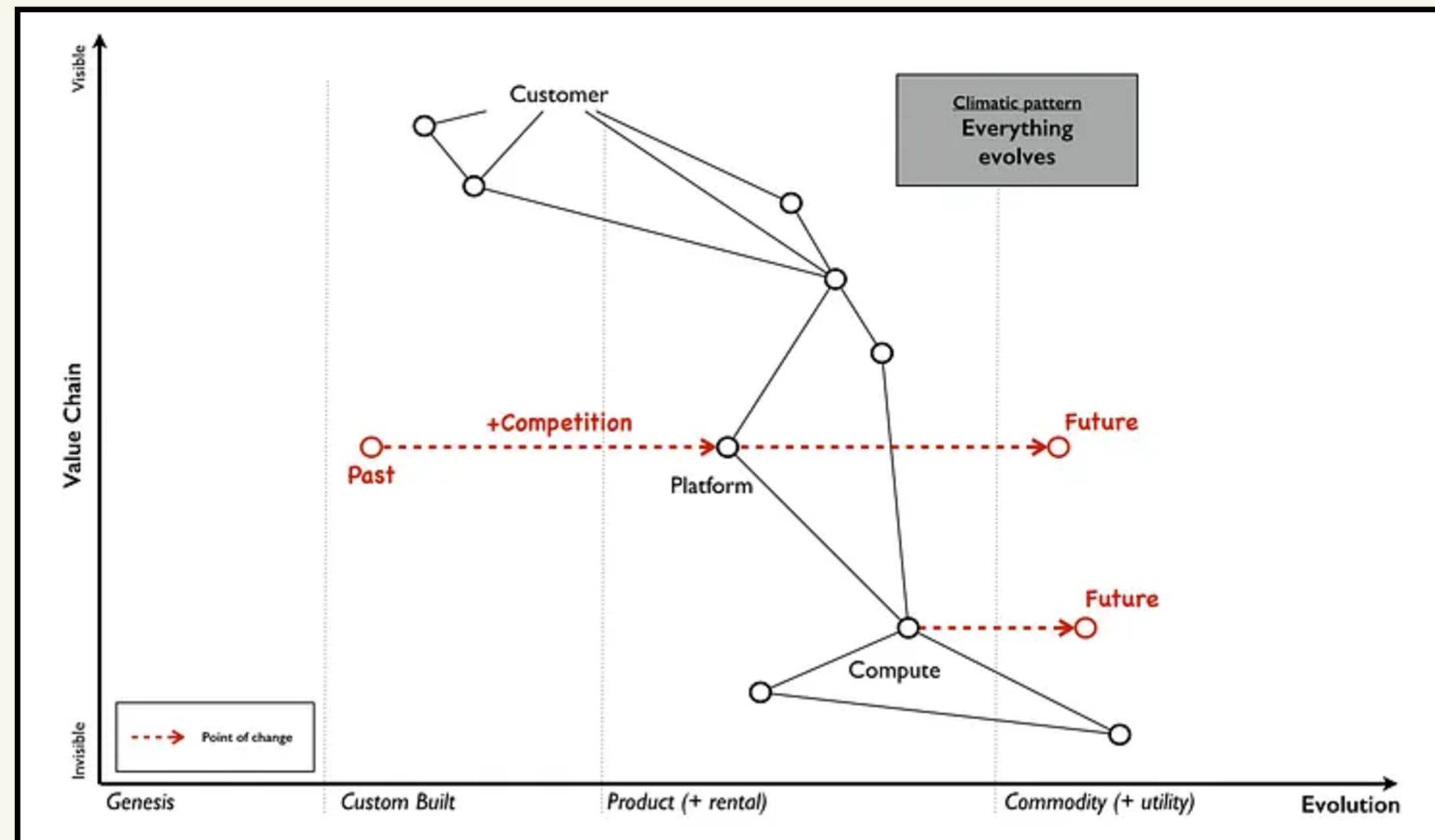
Climate



Climate จะมีรูปแบบเรียกว่า
Climatic Patterns

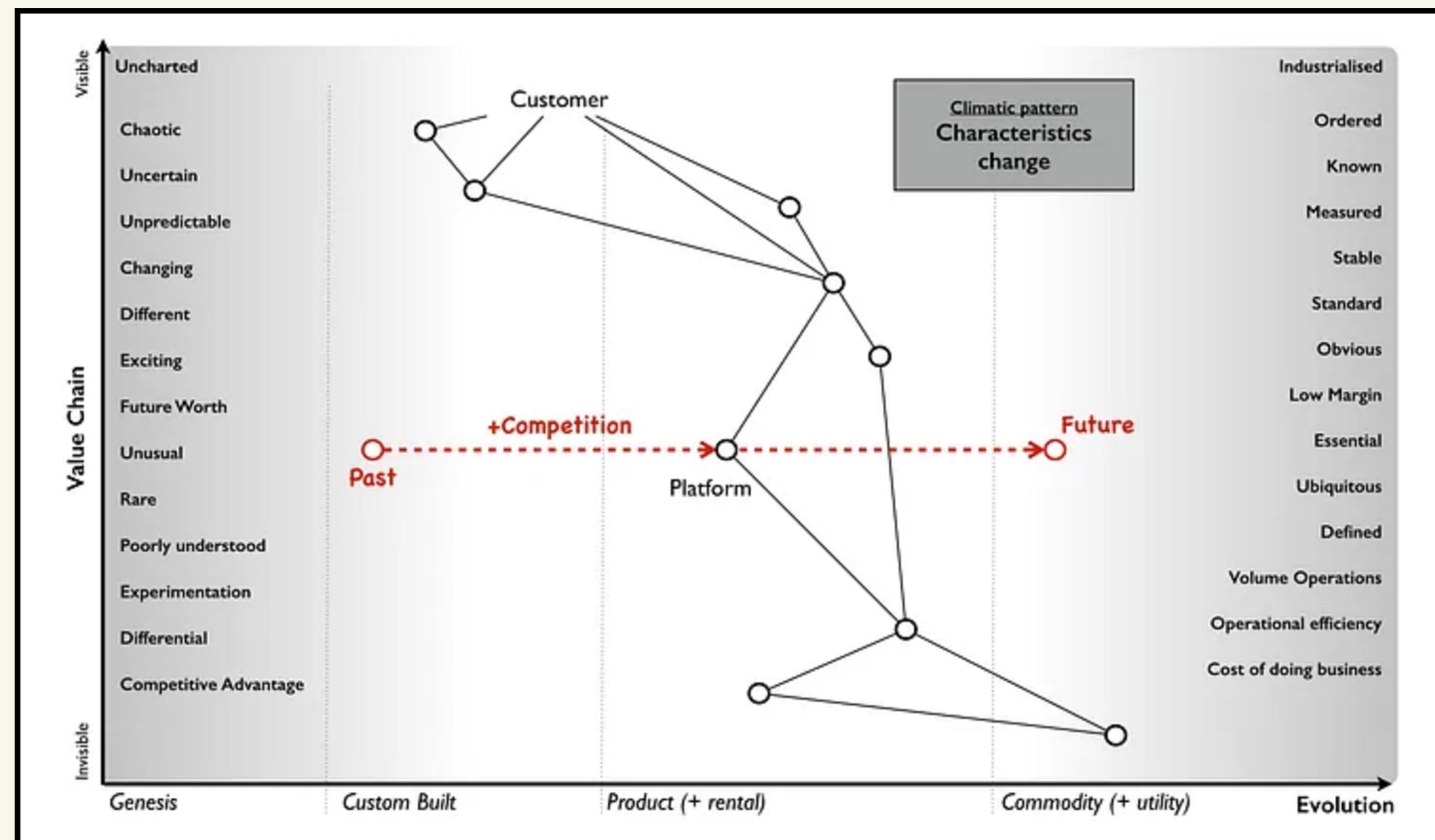
Climatic Patterns

Everything evolves



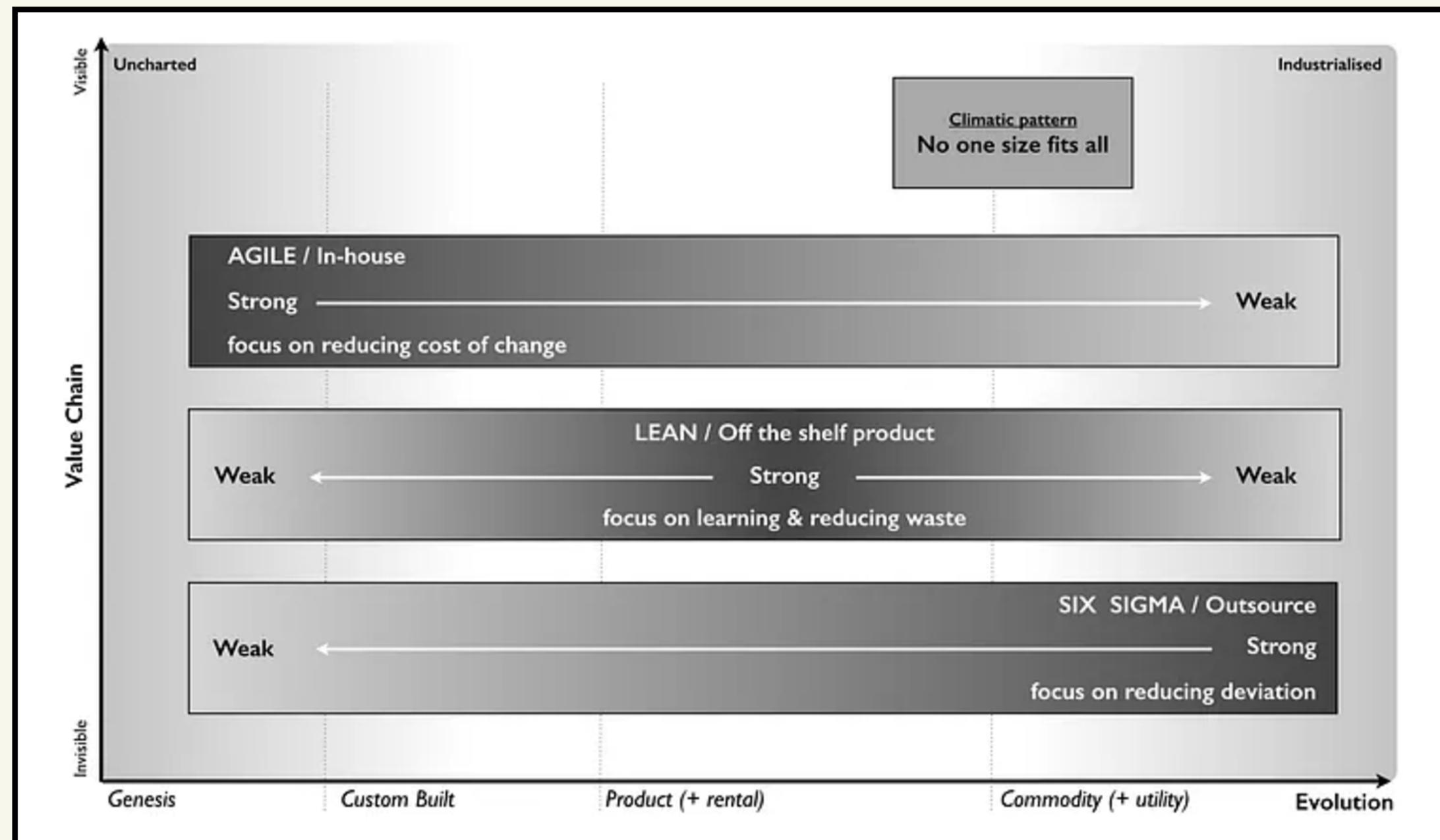
Climatic Patterns

Characteristics change



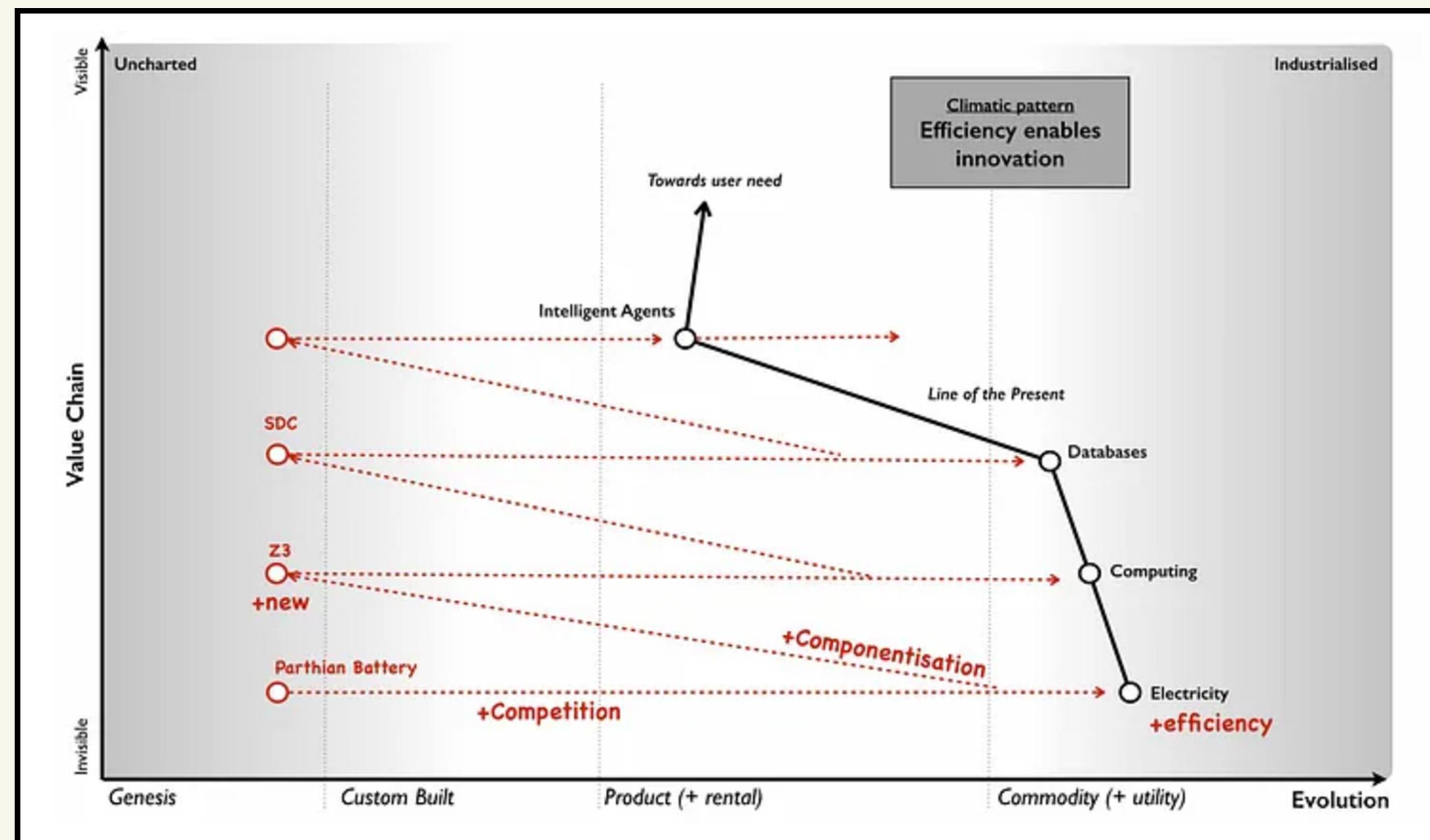
Climatic Patterns

No one size fits all



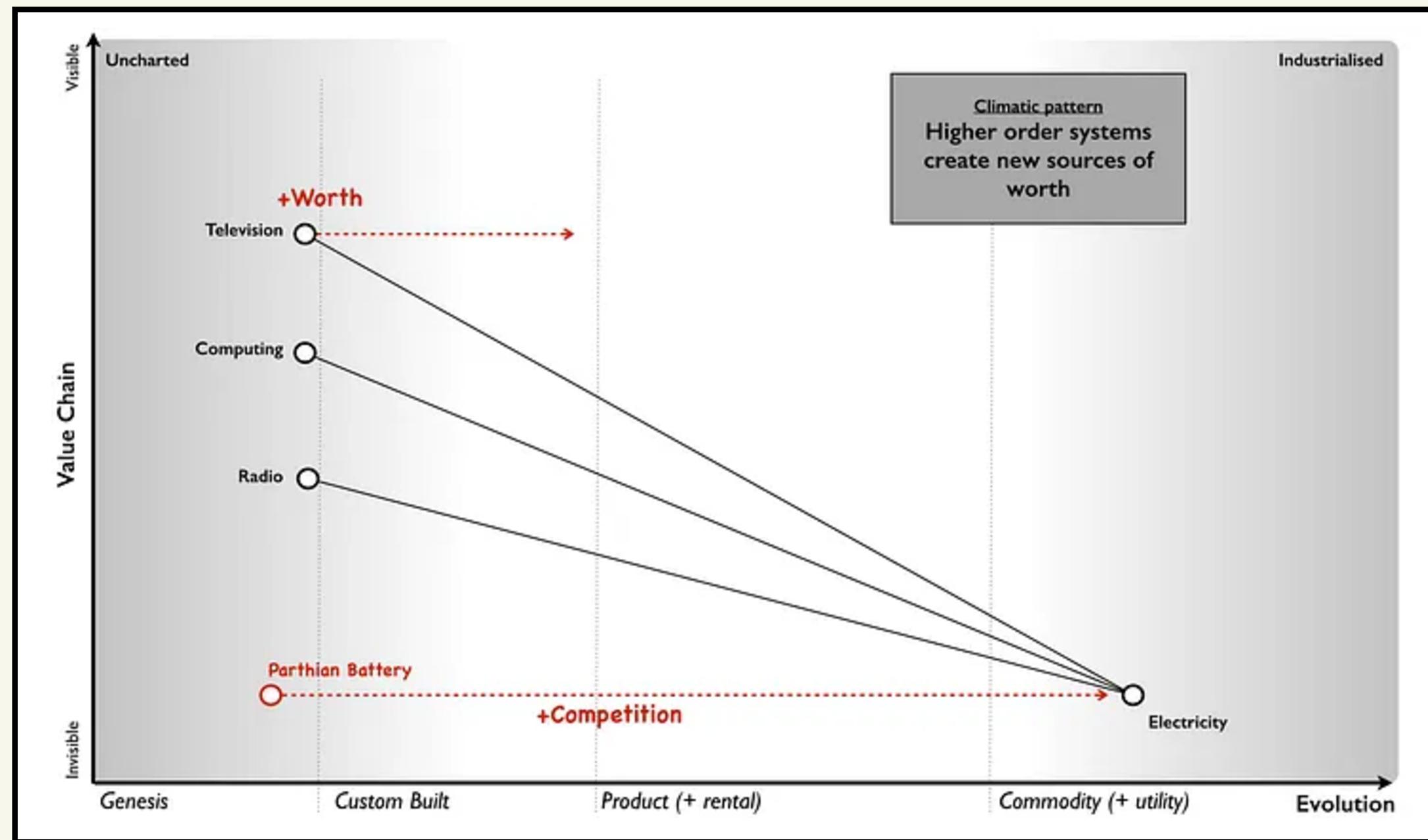
Climatic Patterns

Efficiency enables innovation



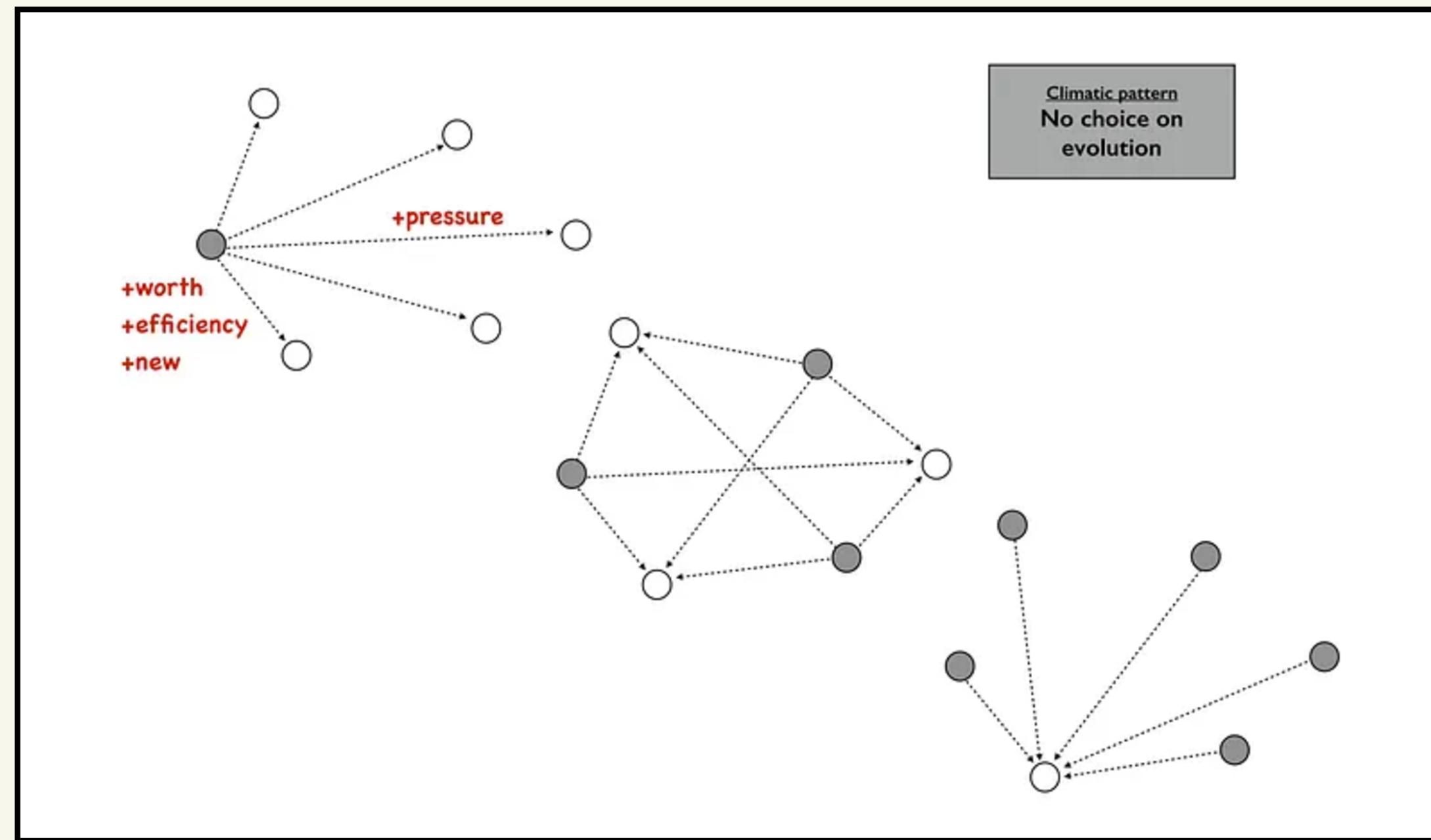
Climatic Patterns

Higher order systems create new sources of worth



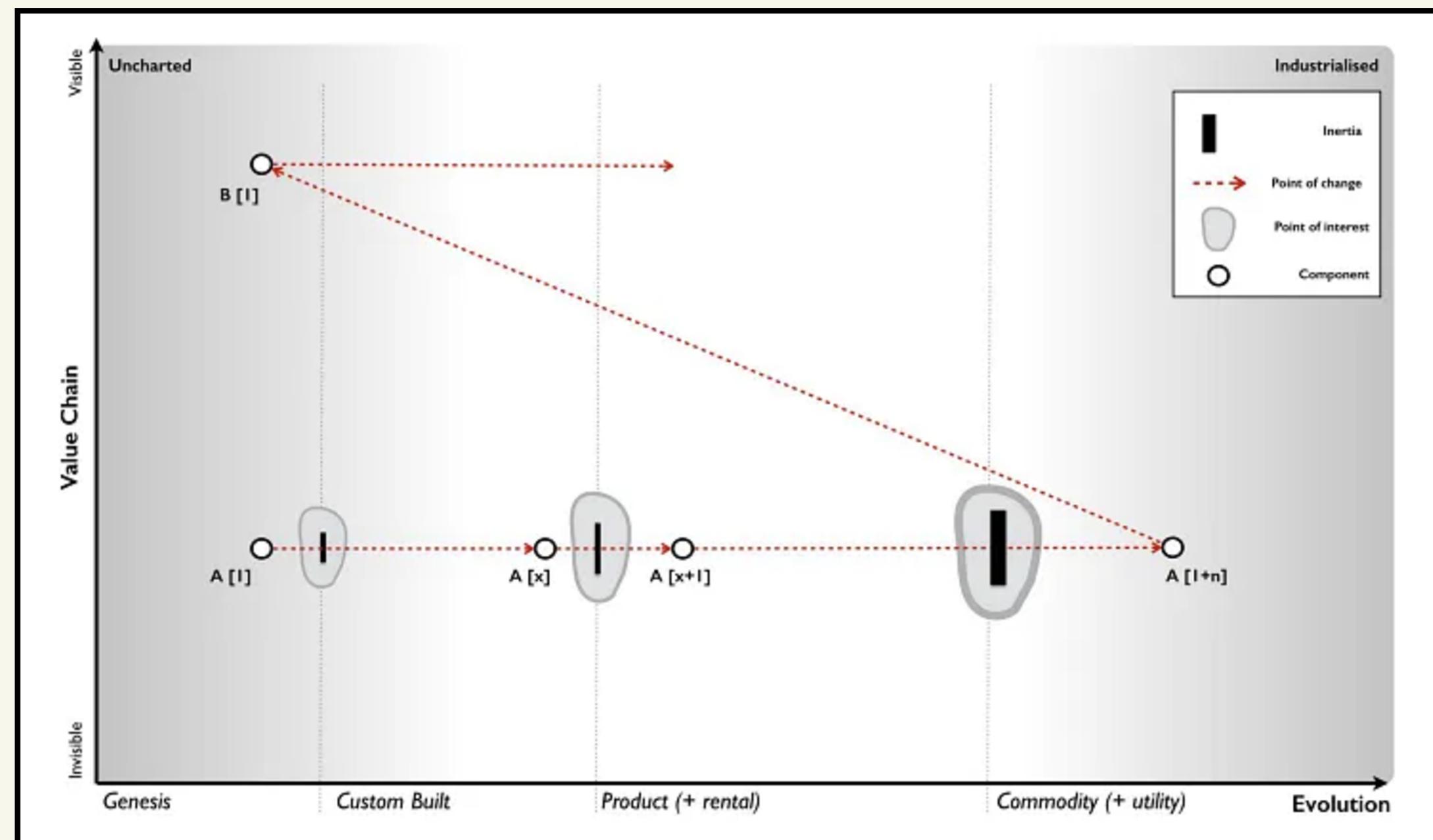
Climatic Patterns

No choice on evolution



Climatic Patterns

Past success breeds inertia



Climatic Pattern

เลือก Climatic Pattern ที่น่าสนใจมากยิ่งกัน
ทำไมถึงสนใจ Climatic Pattern ที่เลือก



เขี่ยบ%



Climatic Pattern

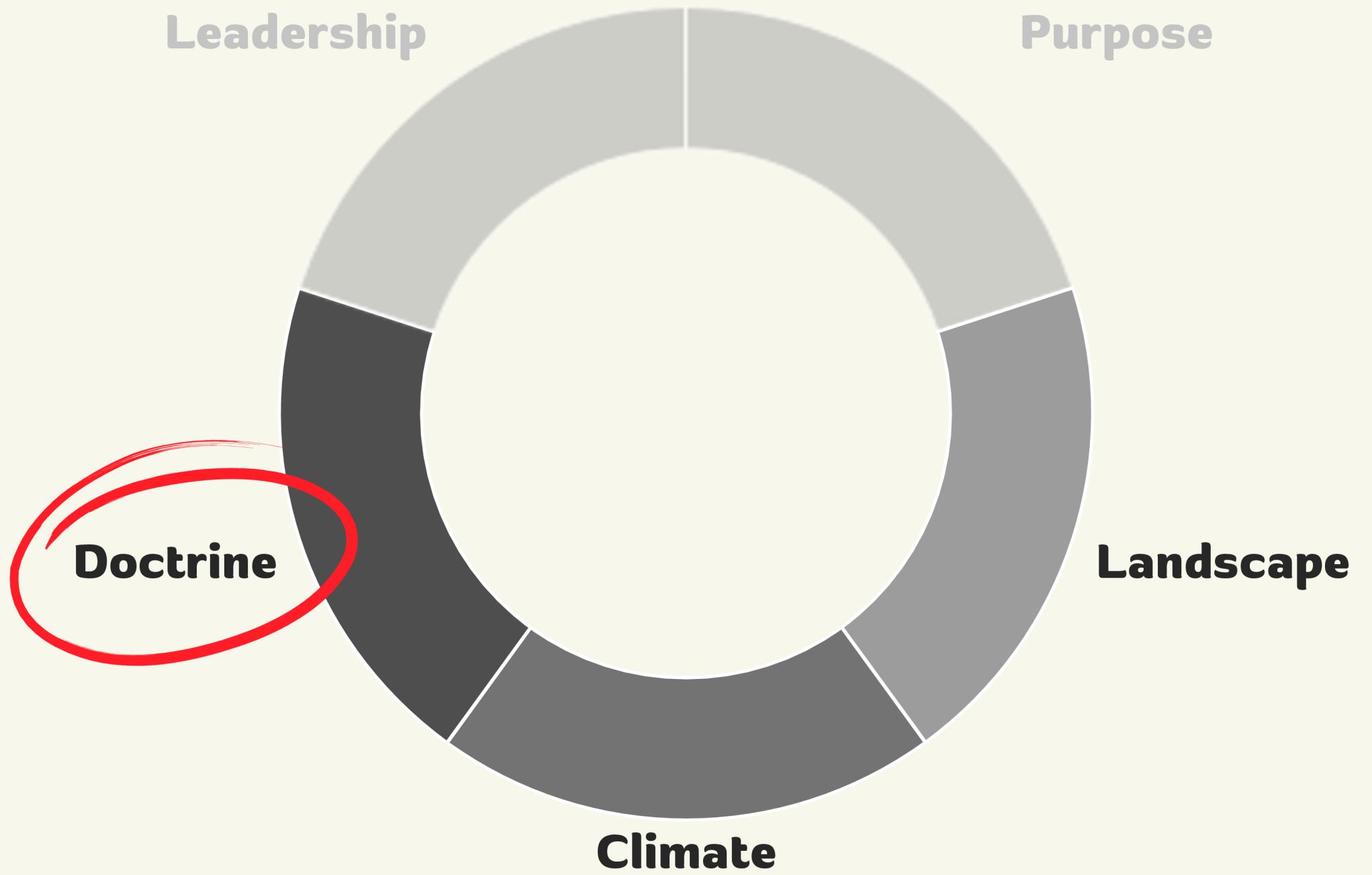
ແນ່ງປັນ Climatic Pattern ທີ່ເລືອກ



ແນ່ງປັນ
ໃຫກລຸມ



Doctrine



แผนการทำงาน ใช้ได้กับทุก
อุตสาหกรรม ไม่ขึ้นกับ Context

Doctrine

Focus on user need

- ตัวเองอยากรีดอะไร  User อยากรีดอะไร
- **User need ก่อน** แล้วค่อยทำกำไร
- วิธีการรับมือที่ต่างกันใน Evolution ที่ต่างกัน
 - **Uncharted** - ห้าง user และ specialist ก็ไม่รู้ว่าต้องการอะไร
 - **Transitional** - ห้าง user และ specialist บอกได้ว่าต้องการอะไร
 - **Industrialized** - ห้าง user และ specialist บอกได้ว่าต้องการอะไร แต่อาจจะมี **bias จาก past success**

Doctrine

Use a common language

- ถึงต่างหน่วยงานกัน ก็ให้ใช้ภาษาเดียวกัน
- ภาษาเดียวกัน รวมถึง **Map** หรือ **Diagram** เดียวกัน

Doctrine

Be transparent

- ใช้ Map เปิดโอกาสให้คนอื่นสามารถเข้าใจ เพื่อทำให้เข้าใจตรงกัน
- ข้อเสียคือคนทำ Map อาจจะไม่สบายใจเวลา มีคนถาม ... !?

Doctrine

Challenge assumptions

- ของใน Map เราอาจจะคิดไปเอง ต้อง~~กล้าให้คนอื่นท้าทาย~~
- การลงโทษ หรือมี bias กับคนที่มาท้าทาย จะทำให้องค์กรเสียหาย

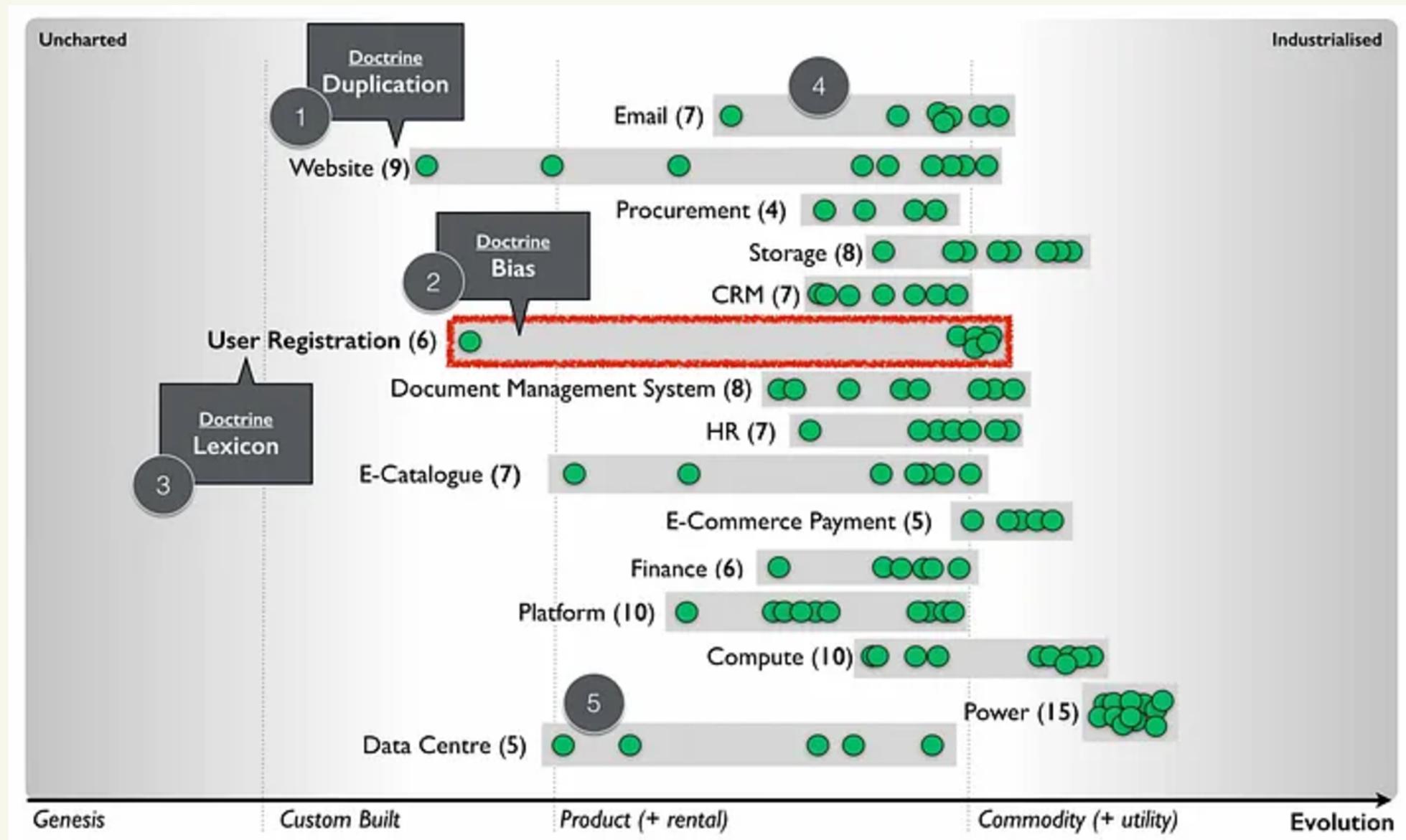
Doctrine

Remove duplication and bias

- ทำ map ไปลักษ์พัก เกิด duplicate component ในหลาย ๆ map อาจจะเป็น component เดียวกัน หรือคนละ component แต่ทำหน้าที่เดียวกัน
- หรือเกิด bias คือ component เดียวกัน แต่บนคนละ map กลับอยู่คนละ stage กัน
- ให้ **ลดความซ้ำซ้อนและ bias** ผ่านนั้นลงโดยการใช้ Map มาพูดคุยกัน อาจจะใช้เครื่องมืออย่างเช่น Profile Diagram หรือ Capability Profile มาช่วย

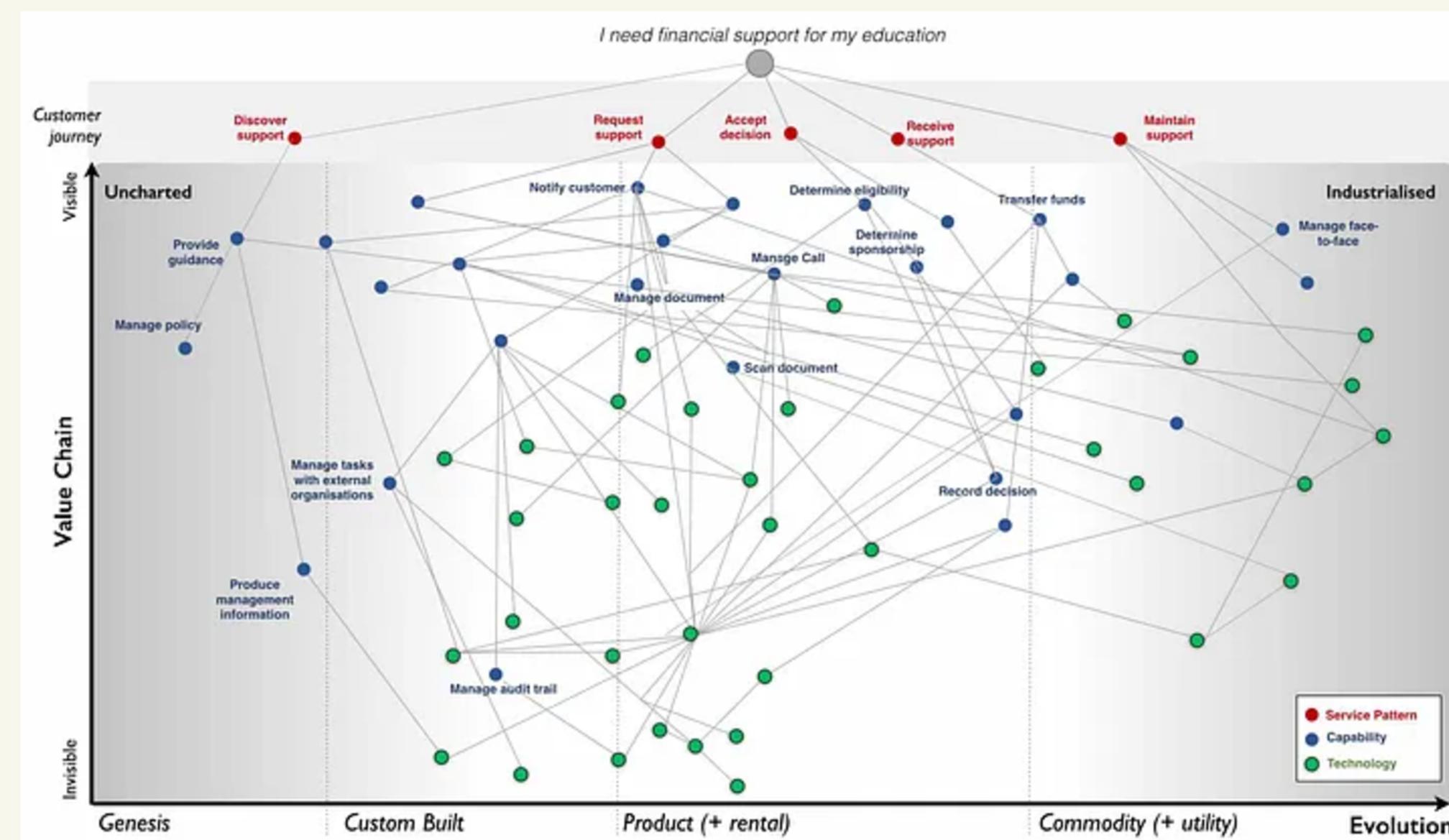
Doctrine

Remove duplication and bias - Profile Diagram



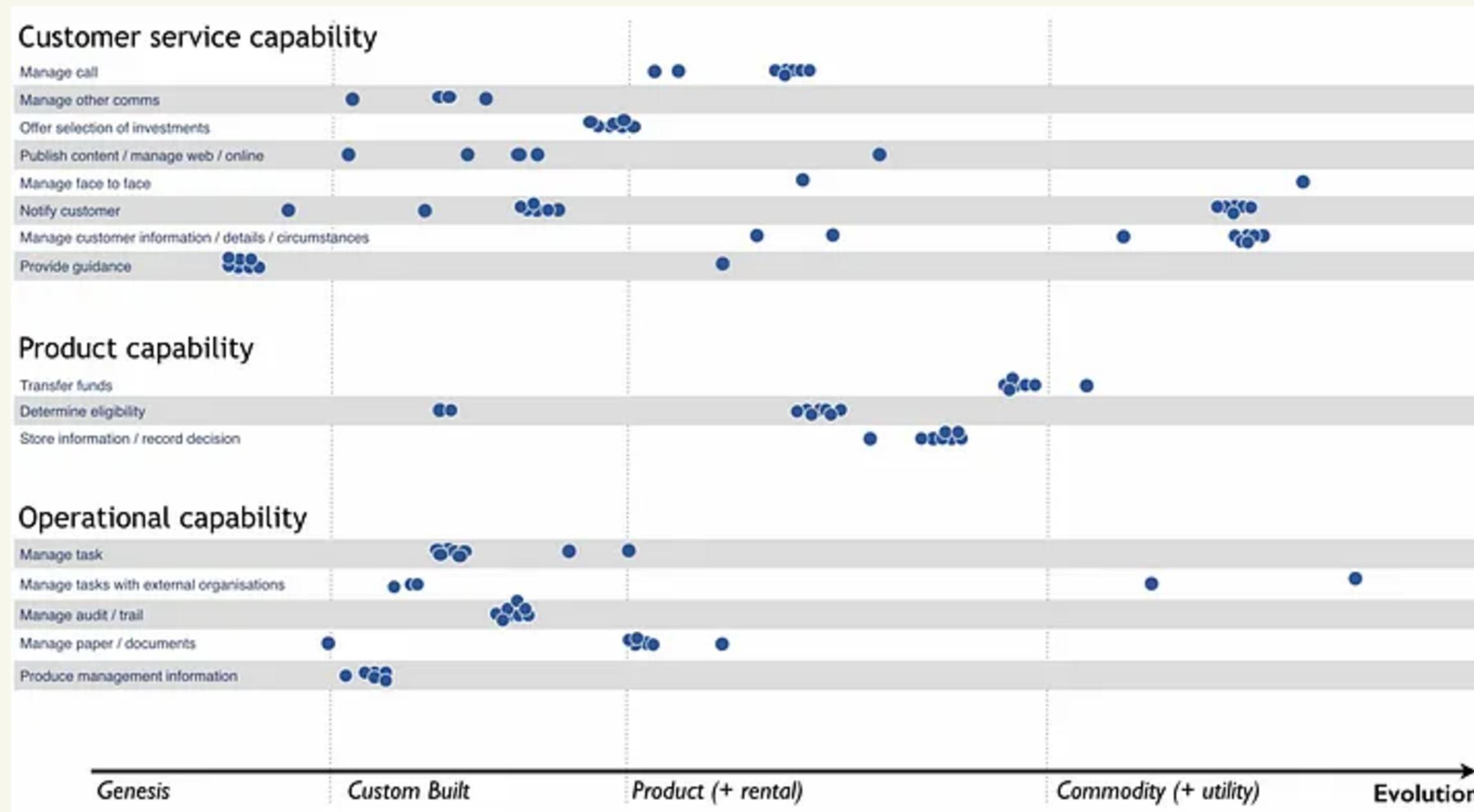
Doctrine

Remove duplication and bias - Map with Customer Journey



Doctrine

Remove duplication and bias - Capability Profile



Doctrine

Use appropriate methods

- ใช้วิธีการที่เหมาะสมสำหรับ component ที่อยู่ Stage ต่าง ๆ กัน
 - Uncharted ใช้ Agile ร่วมกัน in-house
 - Transitional ใช้ Lean ซึ่งของที่มีอยู่แล้วมาใช้
 - Industrialized ใช้ Six Sigma จ้าง outsource

Doctrine

Think small

- แบ่ง Component ให้ย่อย เพื่อให้เห็น Landscape และจัดการได้ง่ายขึ้น
- ใช้ทีมขนาดเล็ก และให้อำนาจตัดสินใจที่มากพอ
 - กำหนดเป้าหมายให้ชัดเจน
 - กำหนด interface กับทีมอื่นให้ชัดเจน
 - กำหนดวิธีการวัดผลการทำงานให้ชัดเจน

Doctrine

Think aptitude and attitude

- Component แต่ละอันต้องการ **ทักษะ (aptitude)** ที่จะทำให้สำเร็จต่างกัน
- Uncharted, Transitional, Industrialized ก็มี **วิธีคิด (attitude)** ที่ต่างกัน
- Pioneers, Settlers, Town Planners
 - **Pioneers** - เน้น Experiment และ Discovery
 - **Settlers** - เปลี่ยน Prototype เป็น Product ทำให้คนอื่นเชื้อใจ มั่นใจ เข้าใจ พึงเลียง ลูกค้า
 - **Town Planners** - Scale การผลิต เน้นประสิทธิภาพ ประหยัด ทำให้ของแพร่หลาย

**สิ่งที่ต้องระวัง คือ มันไม่ใช่การส่งต่องานให้ดูแล
แต่จะเป็นการ build + operate ใน scale / stage ที่ต่างกัน**

Doctrine

Design for constant evolution

- ทำให้องค์กรมี evolution ตลอดเวลา
- Pioneers หา idea ใหม่ ๆ ทดลอง
- Settlers ไปขโมยไอเดียของ Pioneers มาทำเป็น product
- Pioneer แข่งขันโดย หาไอเดียใหม่ ค้นคว้าต่อ
- Town Planners ไปขโมยไอเดียของ Settlers
- Settlers ไปหาไอเดียที่ดีขึ้นจาก Pioneer
- Town Planners ทำ idea ที่ขโมยมา กลายเป็น component ที่เป็น utilities
- Pioneers ไปใช้งาน utilities ที่เกิดขึ้น เพื่อสร้าง idea ใหม่ ๆ

Doctrine

Design for constant evolution

- ถ้าองค์กรมีแค่ Pioneers กับ Town Planners
 - Town Planner รับมือกับความไม่แน่นอนของสิ่งที่ Pioneer ทำได้ ของที่ Pioneer ทำ ก็จะไม่มีวิวัฒนาการ Town Planner ก็จะไปทำแต่ของเดิม ๆ แบบเดิม ๆ
- ตลาดที่ evolve อย่างรวดเร็วอยู่ตลอด product จึงต้อง evolve ตามด้วย
 - Conway's law บอกว่า Product จะ reflect ถึงวิธีการสื่อสารในองค์กร ดังนั้น องค์กร ที่ซับซ้อน ก็ควรจะต้อง evolve วิธีการสื่อสารตามด้วย เพื่อให้ทันกับการ evolve ของ ตลาด

Doctrine

เลือก Doctrine ที่น่าสนใจมาคุยกัน
ทำไมถึงสนใจ Doctrine ที่เลือก
ถ้าไม่ทำ Doctrine ที่เลือกดิด่าว่าจะเกิดอะไรขึ้น (Anti patterns)



เขียน



Doctrine

แบ่งปัน Doctrine ที่เลือก และ Anti pattern



แบ่งปัน
ให้กลุ่ม



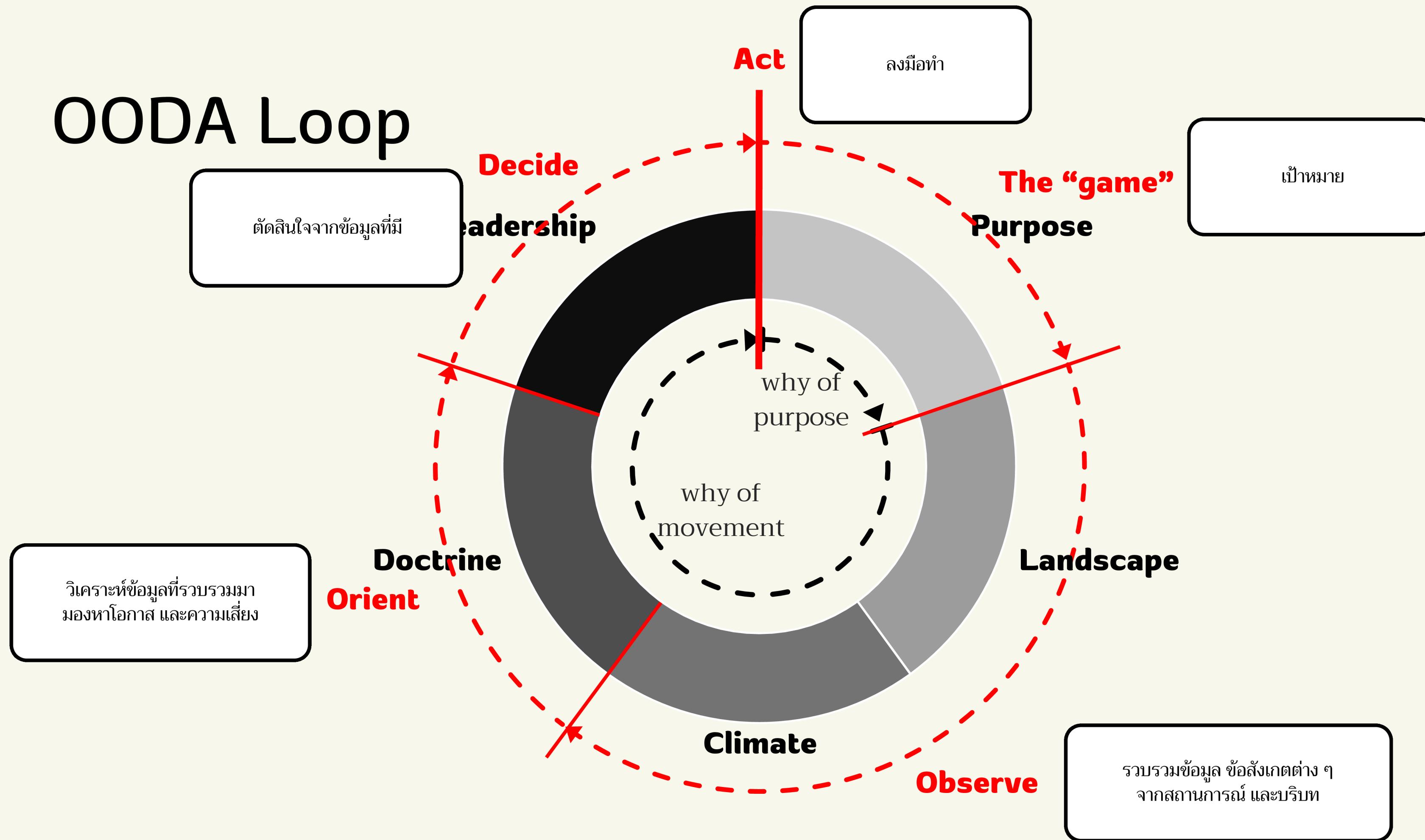
Gameplay

- Strategic action ถูกกระทำตาม context ที่เป็นอยู่
- ทำความเข้าใจ Landscape
- หาจุดที่จะ Evolve อาจจะมากกว่า 1 จุด
- พิจารณา Challenge, Pros & Cons แต่ละจุด
 - บางการตัดสินใจเป็นการเปิดตลาด อย่าพิจารณาแค่ ROI หรือมองว่าเป็น Core business หรือไม่ เหมือนกับการเล่นหماกรุก ที่อาจจะยอมเสียบางตัวเพื่อเป้าหมายที่ยิ่งใหญ่กว่า
- ระวัง Inertia ทำให้ไม่กล้าเสี่ยงที่จะตัดสินใจ
- การตัดสินใจ อาจจะทำให้ Why to purpose เปลี่ยนไปได้

Tips

- Model เป็นแค่ไกด์ มันไม่ถูกต้อง เอาไว้ใช้ Collaborate กัน
- ใช้ Model หลาย ๆ อย่างมาช่วยตอบคำถามบางอย่าง เช่น Financial Model
- สนใจ where ที่จะ attack ก่อน why เพราะ where คือ โอกาส
- ทำซ้ำ และเรียนรู้อย่างต่อเนื่อง เพราะทุกอย่างมัน evolve อย่างต่อเนื่องเช่นกัน
- คิดให้เล็ก เริ่มให้ง่าย อย่ารอ perfect map
- ถ้ามีที่ปรึกษากลยุทธ์ ให้ที่ปรึกษา challenge กลยุทธ์ของเรา ถ้าไม่มี เรียนรู้ซะ

OODA Loop



Anti patterns

ไม่โฟกัสความต้องการของ users

เพราะไม่เห็นความต้องการของ users บางทีก็สับสน เอามาปนกับความต้องการส่วนตัว เช่น ผลกำไร รายได้ ข้อมูล

ใช้ภาษาที่แตกต่างกัน

ใช้เครื่องมือหลายอย่างในการอธิบายเรื่องเดียวกัน บางทีก็ไม่สอดคล้องกัน

Anti patterns

ไม่โปร่งใส

บางทีก็ทำให้มองไม่เห็นเรื่องบางอย่าง เช่น คำถานที่ถามว่า เรา มีโปรเจค IoT กีอัน เพราะว่ามันซ่อนอยู่ใน silo

ไม่เคยถูก challenge

หลาย ๆ การตัดสินใจเกิดขึ้นตาม meme หรือ Hippo หรือตามบทความดัง ๆ ใน HBR ซึ่งหลาย ๆ ครั้งมันก็ไม่เวิร์ค

Anti patterns

ไม่สามารถลดความช้าช้อนหรือ bias ลงได้

เพราะไม่ปะรัง Lis ไม่โฟกัส ก็ทำให้ไม่เห็นอะไร และทำให้เกิดแรงต่อต้านในการเปลี่ยนแปลง

ไม่สามารถใช้วิธีการที่เหมาะสมได้

เช่น ใช้ outsource ทุกโครงการ หรือใช้ Agile ทั้งหมด

Anti patterns

คิดเล็กไม่ได้

ทำทีก็จะใหญ่ ต้องลังการ เช่น การทำ platform re-engineering หรือ re-organization ครั้งใหญ่

คิดแบบ aptitude และ attitude ไม่ได้

มองเป็นแค่แบบเดียว (แบบใดแบบหนึ่ง) หรือมองแค่ว่ามีแต่ IT หรือส่งไป train general course ที่ครอบคลุมทุกอย่าง เช่น ส่งทุกคนไปเรียน Agile

Anti patterns

คิดเพื่อให้เกิด evolution ไม่ได้

แบ่ง Organization ตาม meme ใหม่ ๆ ที่โผล่มา เช่น Cloud Department, Digital Department, Big Data Group ก่อให้เกิด Bi-modal IT concept คือ แบ่ง IT เป็น 2 กลุ่ม กลุ่มที่เน้นเรื่องใหม่ ๆ กับกลุ่มที่เป็น Core operation ขององค์กร

Pioneers VS Town Planners

Anti patterns

ไม่สามารถทำให้ Purpose, Mastery และ Autonomy เกิดขึ้นจริงได้

บางทีก็ทำให้ purpose ไปปนกับความรู้สึกที่ขาดการควบคุม และไม่สามารถ influence ได้

ไม่เข้าใจ Basic economic pattern

อยากรצהทำหิ้ง innovation และ efficiency หิ้งคู่ แต่ไม่รู้เรื่องของความสัมพันธ์ของหิ้งสองอย่าง หรือคิดว่ามีทางเลือกให้เปลี่ยน (เช่น จะใช้ Cloud) หิ้ง ๆ ที่ความเป็นจริงแล้วไม่มีทางเลือกเลย หรือติดอยู่กับความสำเร็จในอดีต จนไม่รู้ตัวถึง inertia

Anti patterns

ไม่เข้าใจ context specific play

ไม่ได้ใช้คำตามบริบท แต่กลับไปใช้คำที่เป็น meme แทน เช่น open source, innovation โดยที่ไม่ได้เข้าใจว่าอะไรเหมาะสมกับแบบไหน

ไม่เข้าใจ Landscape

ไม่เข้าใจ component ต่าง ๆ อย่างแท้จริง และไม่เข้าใจความซับซ้อนที่มีอยู่ในองค์กร ทำให้ไม่สามารถอธิบาย capability ที่มีอยู่ได้

Anti patterns

ไม่เข้าใจ strategy

รู้แค่ว่า strategy คือ why แต่แยกไม่ออกระหว่าง why of purpose กับ why of movement รวมถึงคุยกันน้อยมากถึง position กับ movement และยังไม่สามารถอธิบายได้ว่าควรจะต้อง attack ตรงไหนต่อ หรือไม่รู้แม้กระทั่งความสำคัญของ where มากกว่า why

Make a move

วิวัฒนาการ Component
ใน Wardley Map ที่ทำไว้
พิจารณาถึง Climatic Pattern ที่อาจจะเกิดขึ้น
และ Doctrine ที่จะใช้



เขี่ยบ%



Make a move

แบ่งปัน Strategy ที่เลือก



แบ่งปัน
ในกลุ่ม

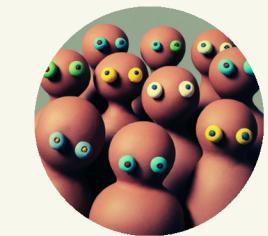


สรุป

Wardley Map และการนำไปใช้งาน



แนะนำ
แผนปักธงของ



Materials



ตารางประกอบ



สไลด์

ตาม/ตอบ