Information Design Patterns for Resilience and Sustainability

Shahryar Eivazzadeh
BTH University
May 2012

Design Patterns

http://www.amcofdata.com/images/Red-Cups.jpg

How we go from here …

- (**#Q**) Questions
- (**#L**) Literature Review
- (**#M**) Information Dynamics Models
- (**#P**) Patterns
  - (**#R**) Impact on Resilience
- (**#A**) Application
  - Simulation of Application

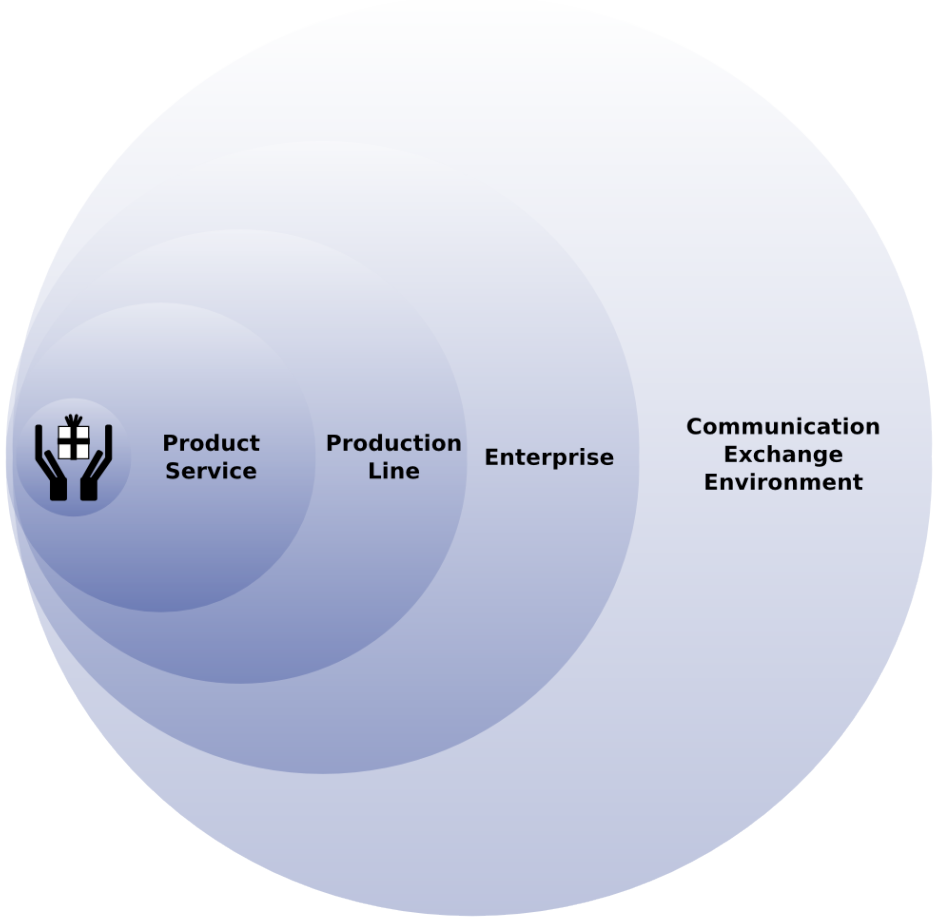
Questions

#Q Questions

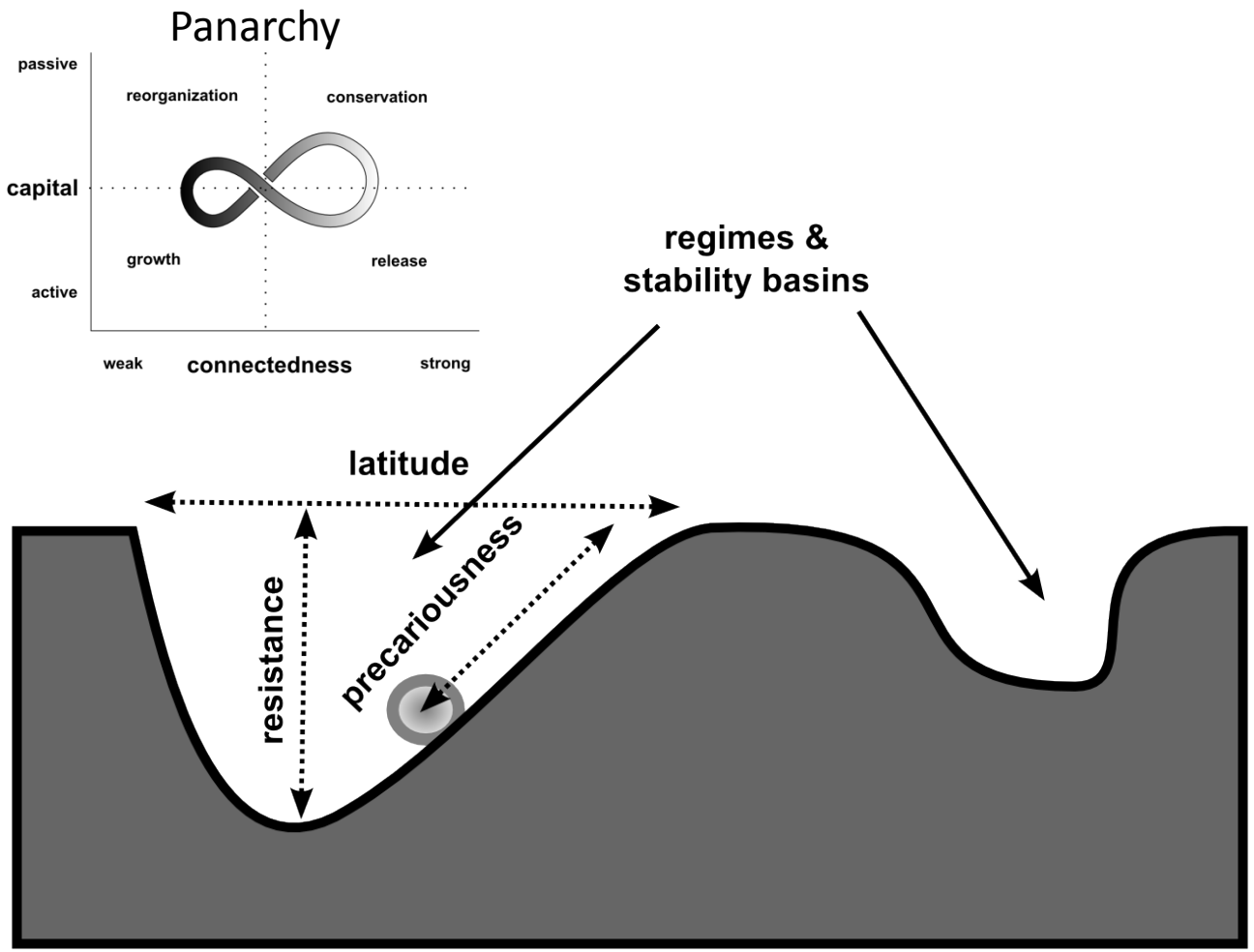
- Is there any (and what are the) **information pattern**(s) that can make systems (especially **product-services**) more **resilient and sustainable**?
- can we organize them as **design patterns** for later reuse in design process?

Literatures

#L Product Service Systems Extended



#L Resilience Theory



#L Resilience and Sustainability

#L Shannon Information Theory

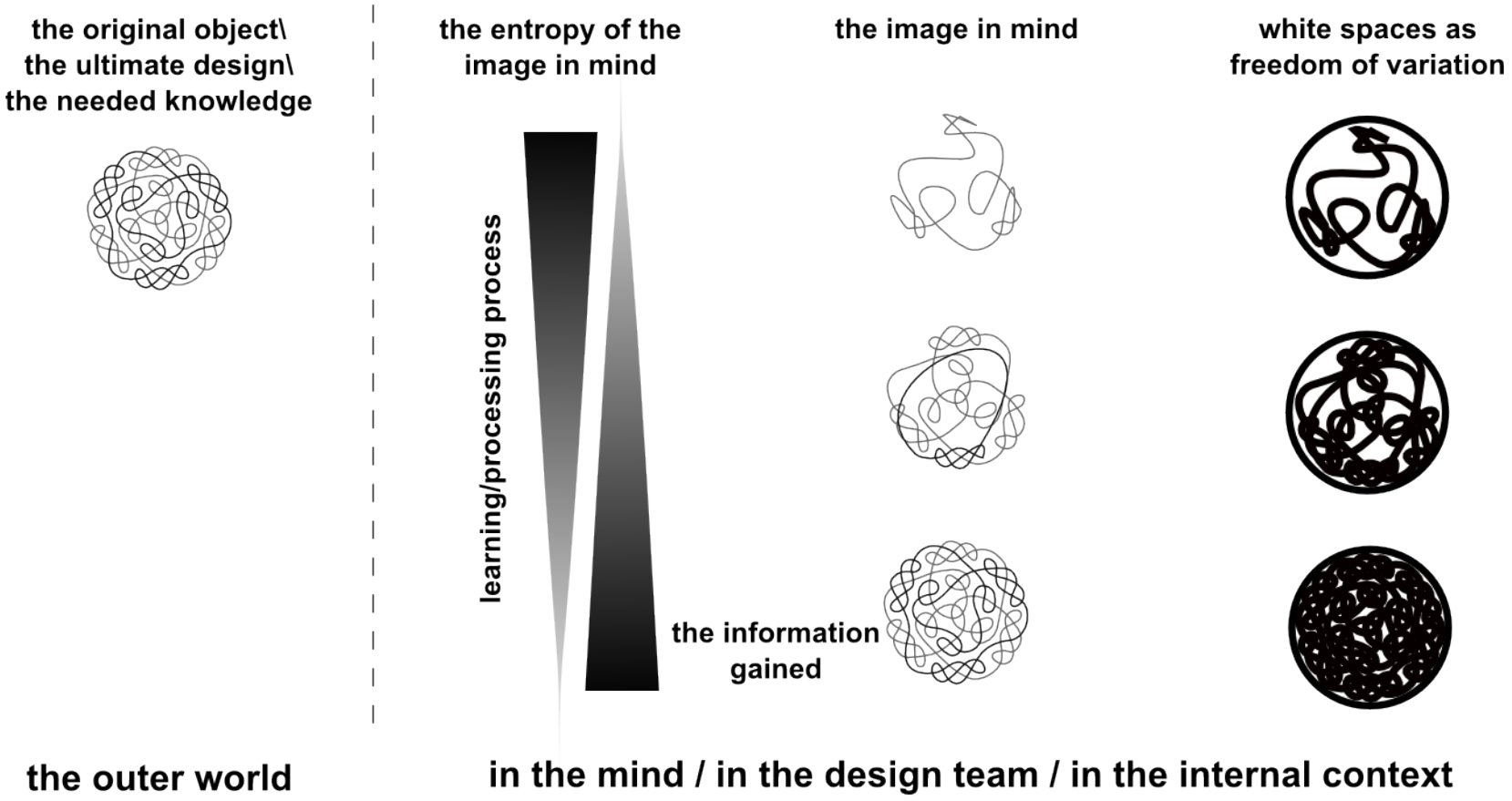
$$Boltzmann\ \ S(x) = k\sum_{i=1}^{n} log(W)$$

$$I(x) = \sum_{i=1}^{n} log(\frac{1}{p(x_i)})$$

$$H(x) = \sum_{i=1}^{n} p(x_i)I(x_i) = \sum_{i=1}^{n} p(x_i)log(\frac{1}{p(x_i)})$$

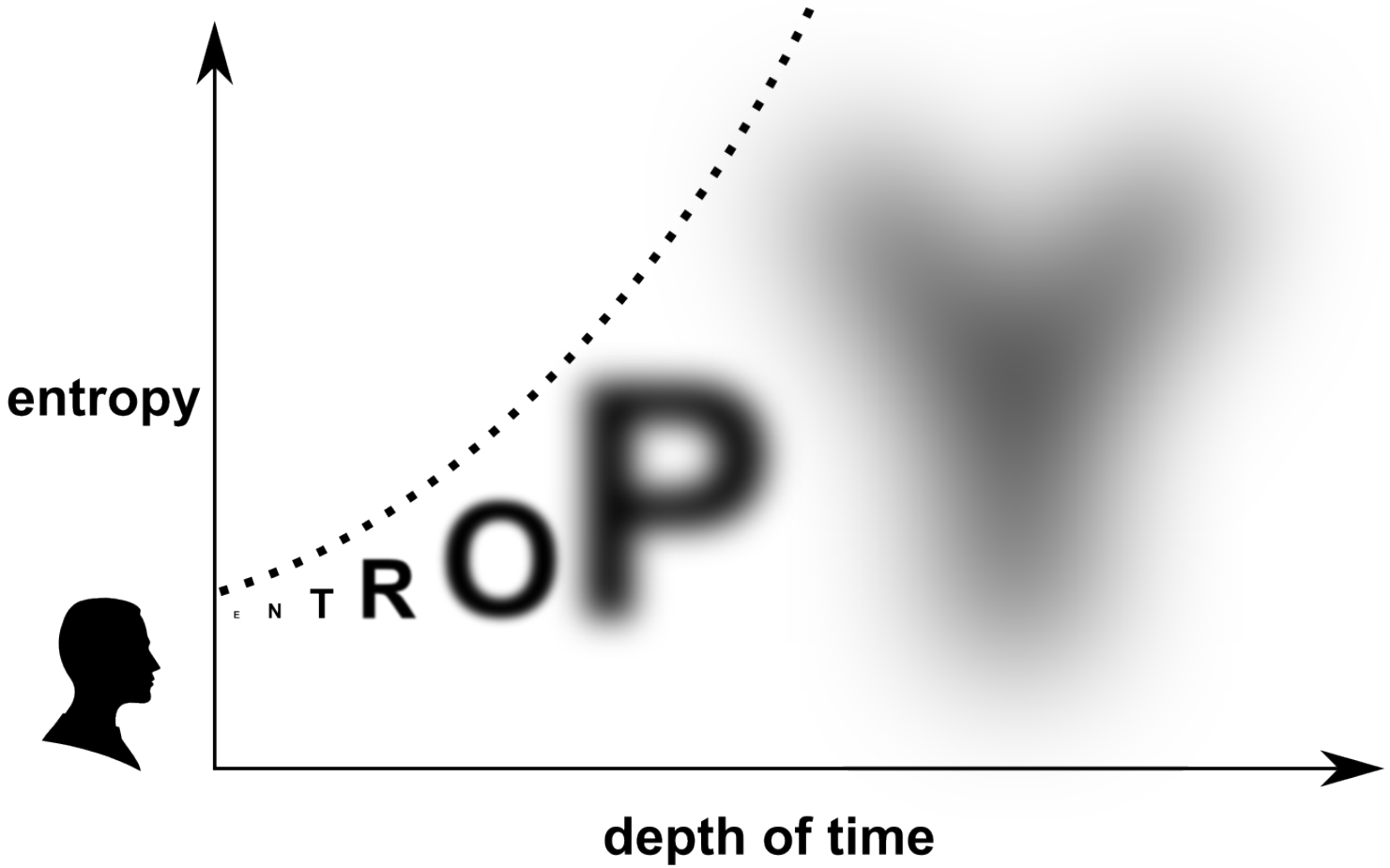
#L Shannon Information Theory

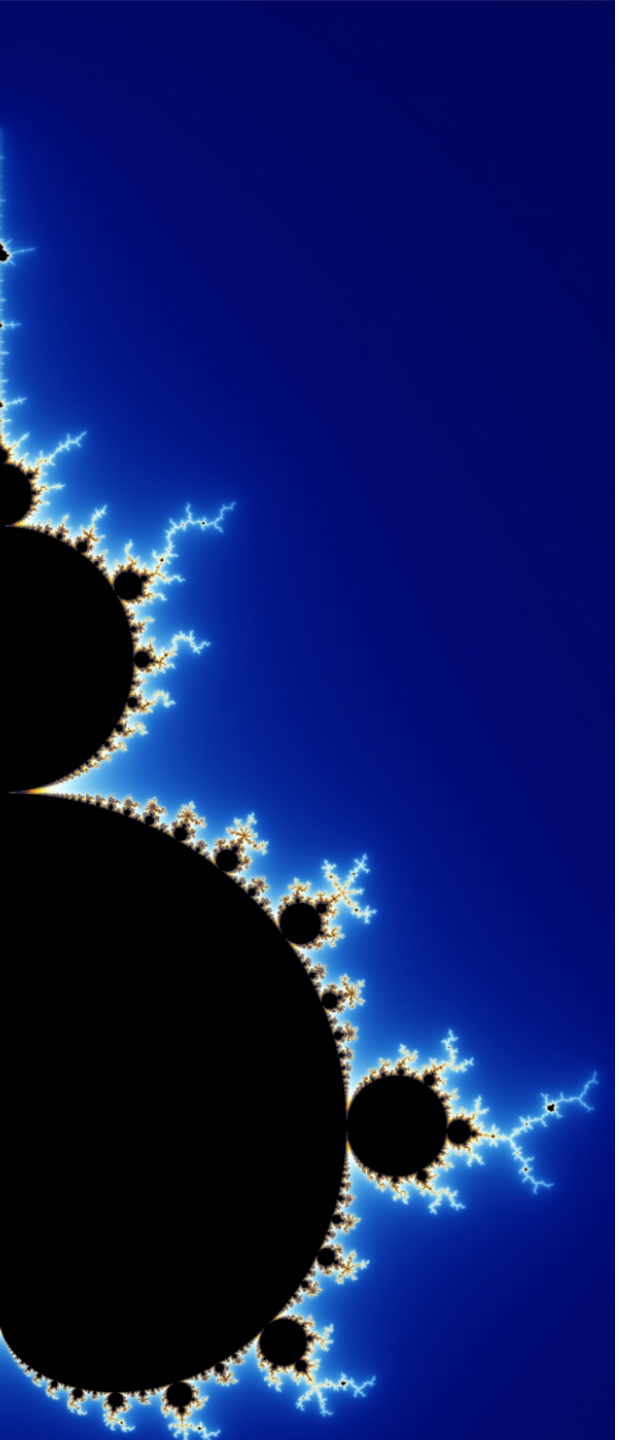
From Which Perspective?



(Eivazzadeh 2012)

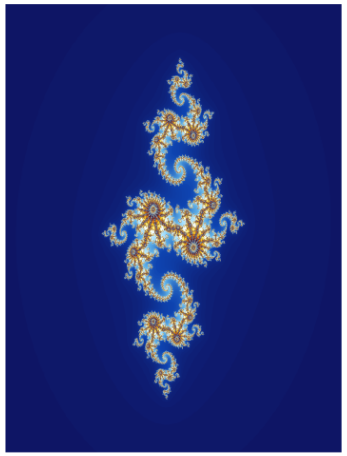
#L Shannon Information Theory
Entropy of Future





Mendelbrot Set, Wolfgang Beyer, by Ultra Fractal3

#L Algorithmic Information Size



Mendelbrot Set, Wolfgang Beyer, by Ultra Fractal3

Yves Guillou Dragon

$Z = Z^2 + c$ & code

~ 4,000 bits

svg image

~ 40,000 bits

svg image

∞

svg image

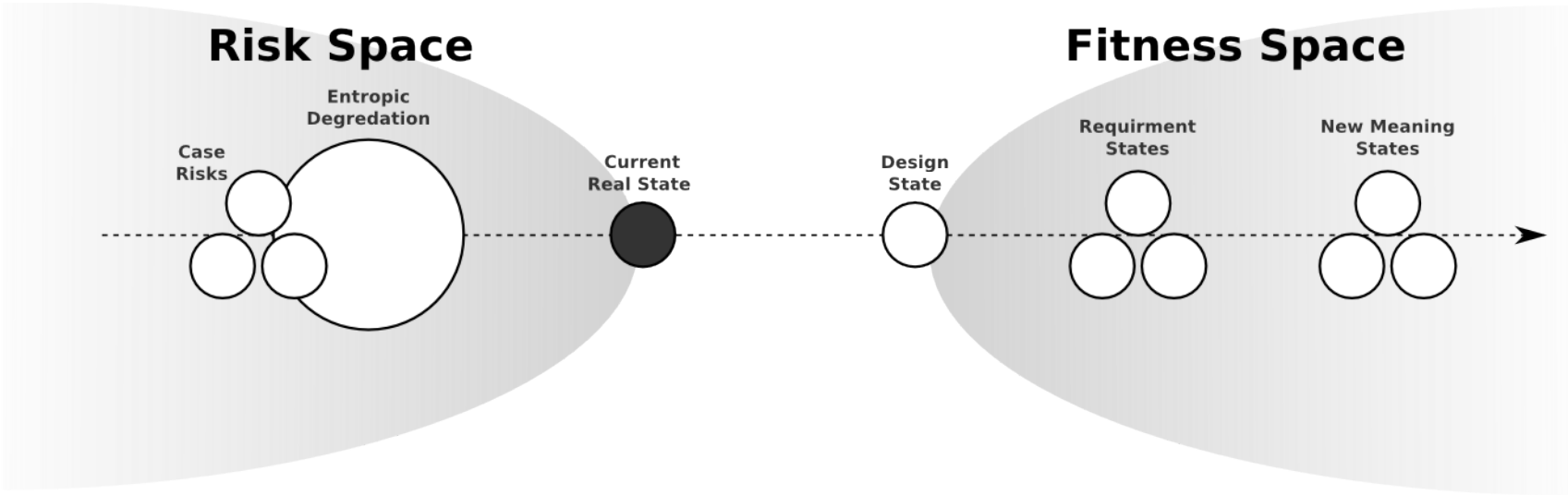
~ 40,000 bits

#L Resilient Systems Models

- **Viable Systems Model (Beer)**
  - System 1: Operation
  - System 2: Network
  - System 3: Rules & Policies
  - System 4: Looking to Future
  - System 5: Balance between the (1,2,3) and (4)
  - Recursion
- **Dynamic Capabilities Model (Teece)**
  - Sensing
  - Seizing
  - Transforming/Managing Treats/Opportunities
- **Living Systems Theory (information subsystems) (Miller)**
  - Input Transducer
  - Internal Transducer
  - Channel and Net
  - Timer
  - Decoder
  - Associator
  - Memory
  - Decider
  - Encoder
  - Output Transducer

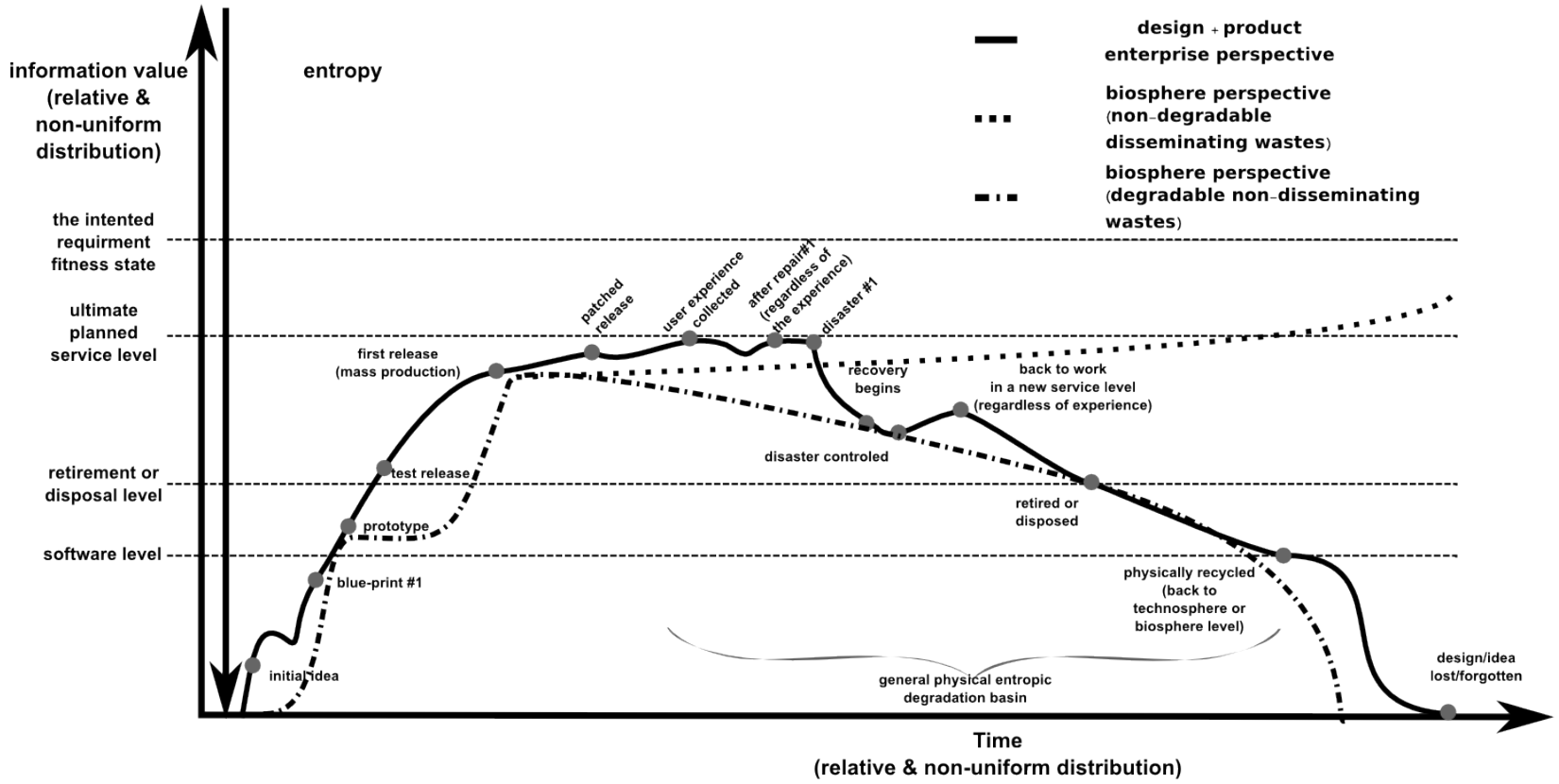
Models

#M Dynamic State-Diagram of a Product Service System



(Eivazzadeh 2012)

#M Information Life Cycle of a Product-Service System



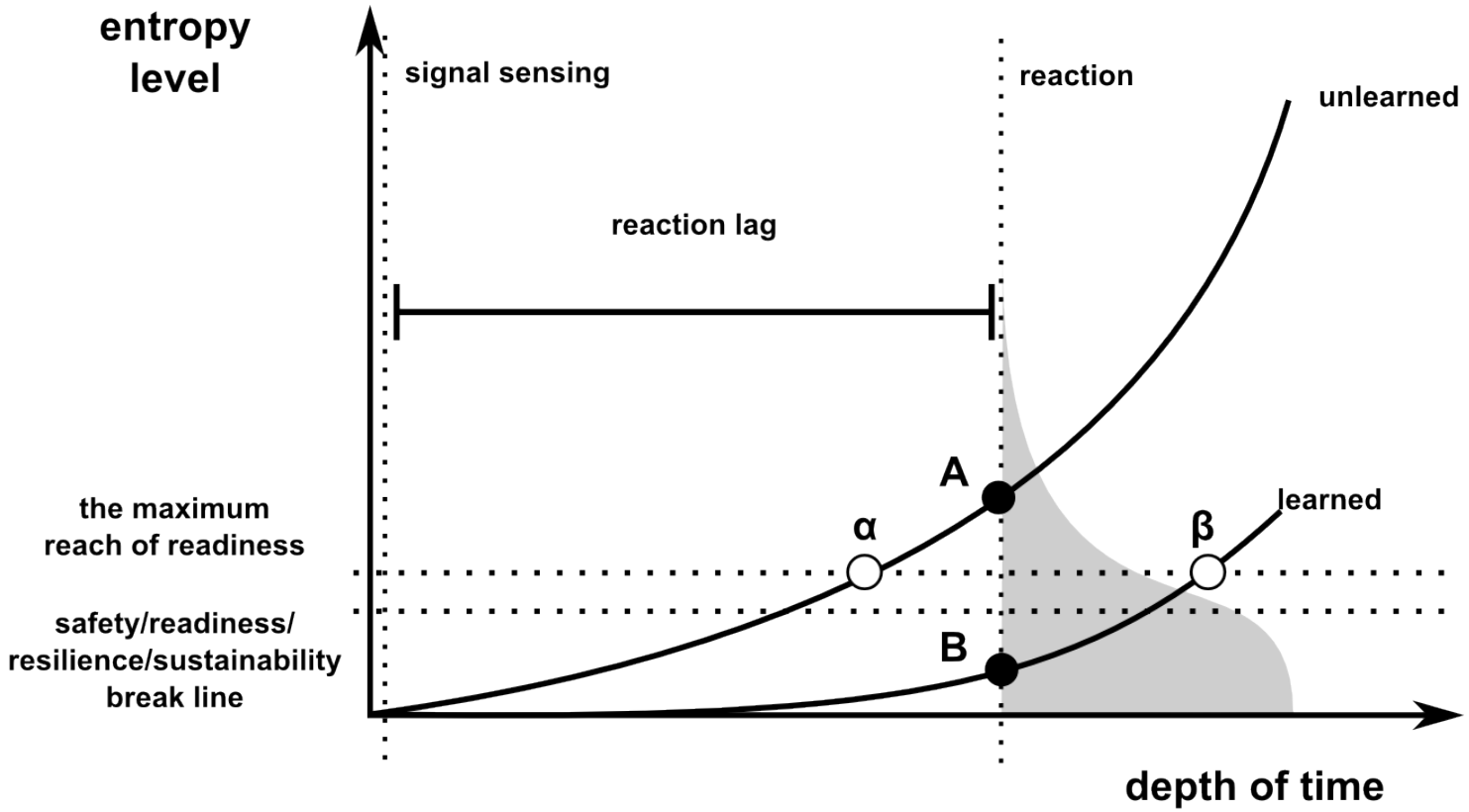
(Eivazzadeh 2012)

Patterns

#P Patterns

- **Fundamental**
- **Tactical**
- **Structural**

#P LSR Triad

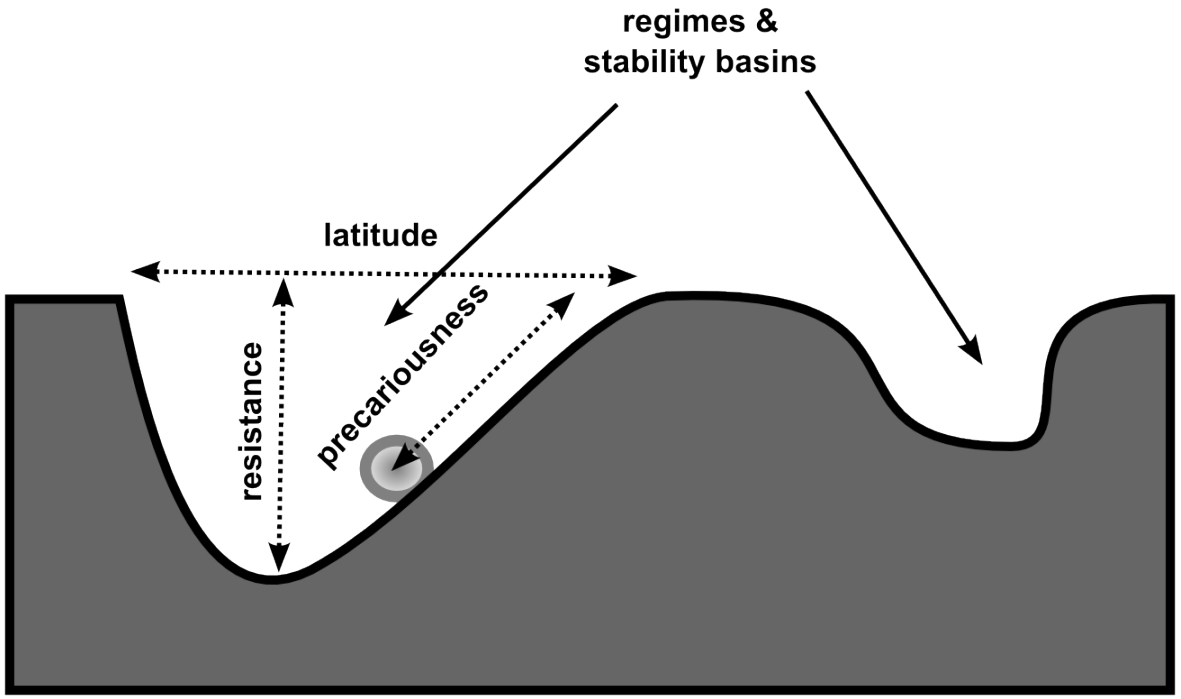


$$E_{environment}(t_i + max[T_{reaction\ lag}]) <= (1 - smax_{safe}[R_{risk}(e_j)]) * E_{max\ entropy}$$

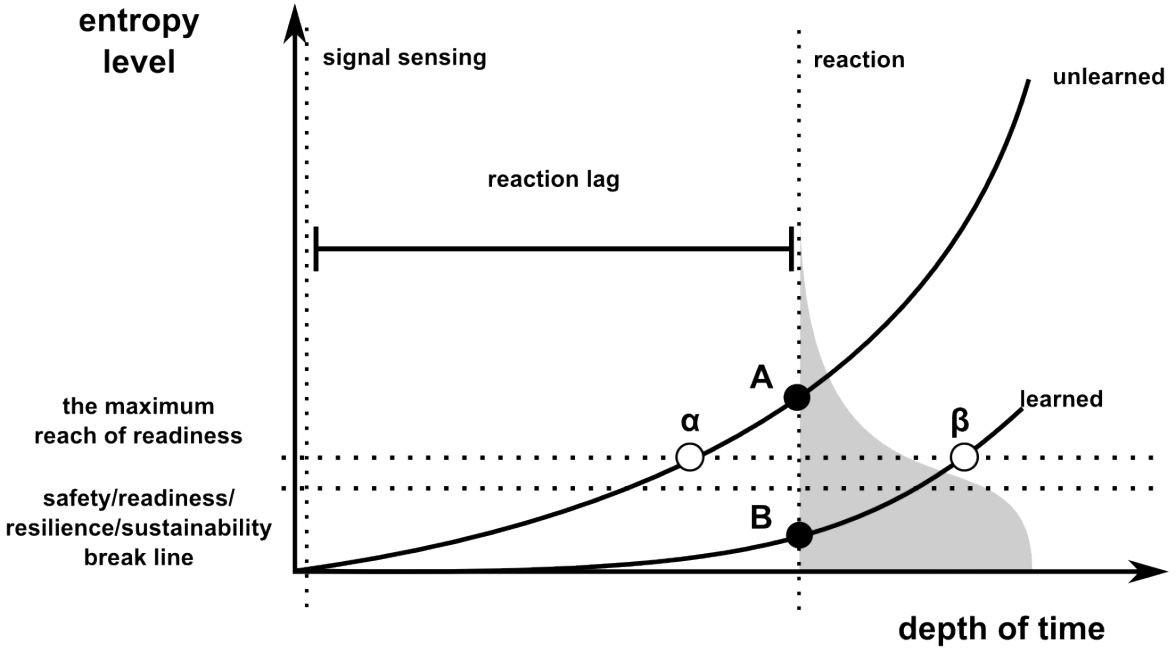
(Eivazzadeh 2012)

#R LSR Triad

- **Learning** the future (less entropic) : increase in ***Latitude***
- **Simpler** future (less entropic) : increase in ***Latitude***
- Faster the **Response** (can avoid the in sight risks): increase in ***Resistance***



#P Learning for Future vs. On-Time Processing Balance



$$p_i(\theta) = c_i + \frac{1 - c_i}{1 + e^{-a_i(\theta - b_i)}} \Rightarrow E_{entropy} \propto e^{b(t+\Delta t) - \theta(t+\Delta t)}$$

Item Response Theory (Baker 2001) (Eivazzadeh 2012)

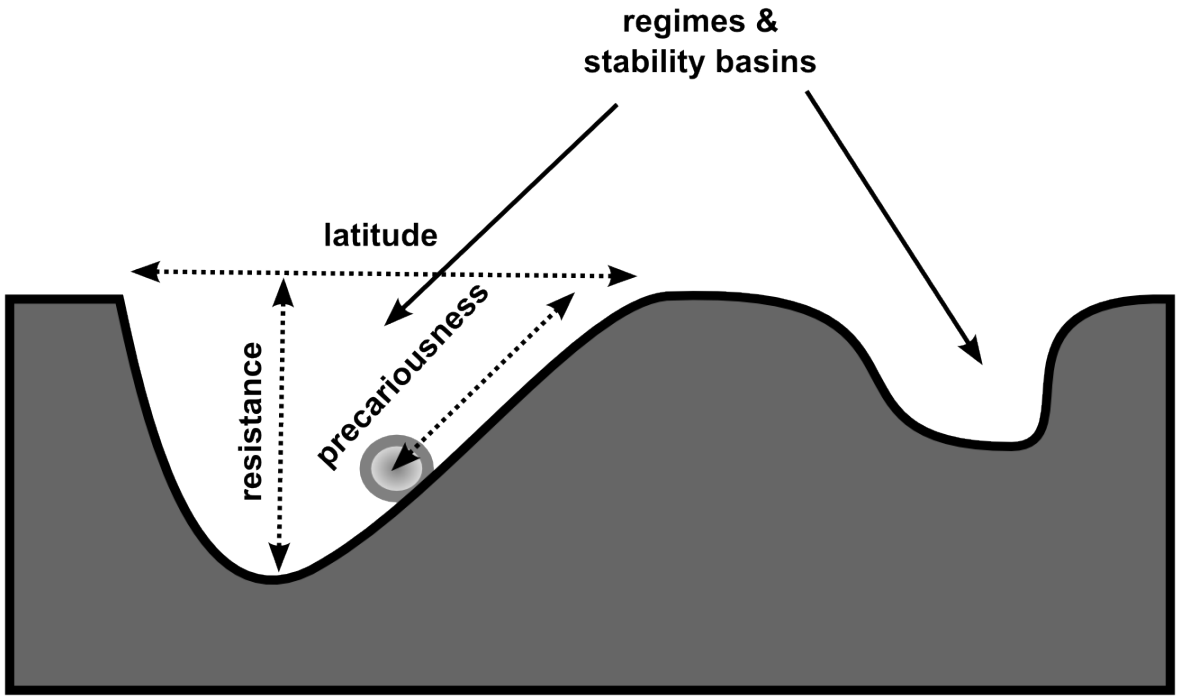
Bend Down VS Level Up the Readiness

#P Learning for Future vs. On-Time Processing Balance

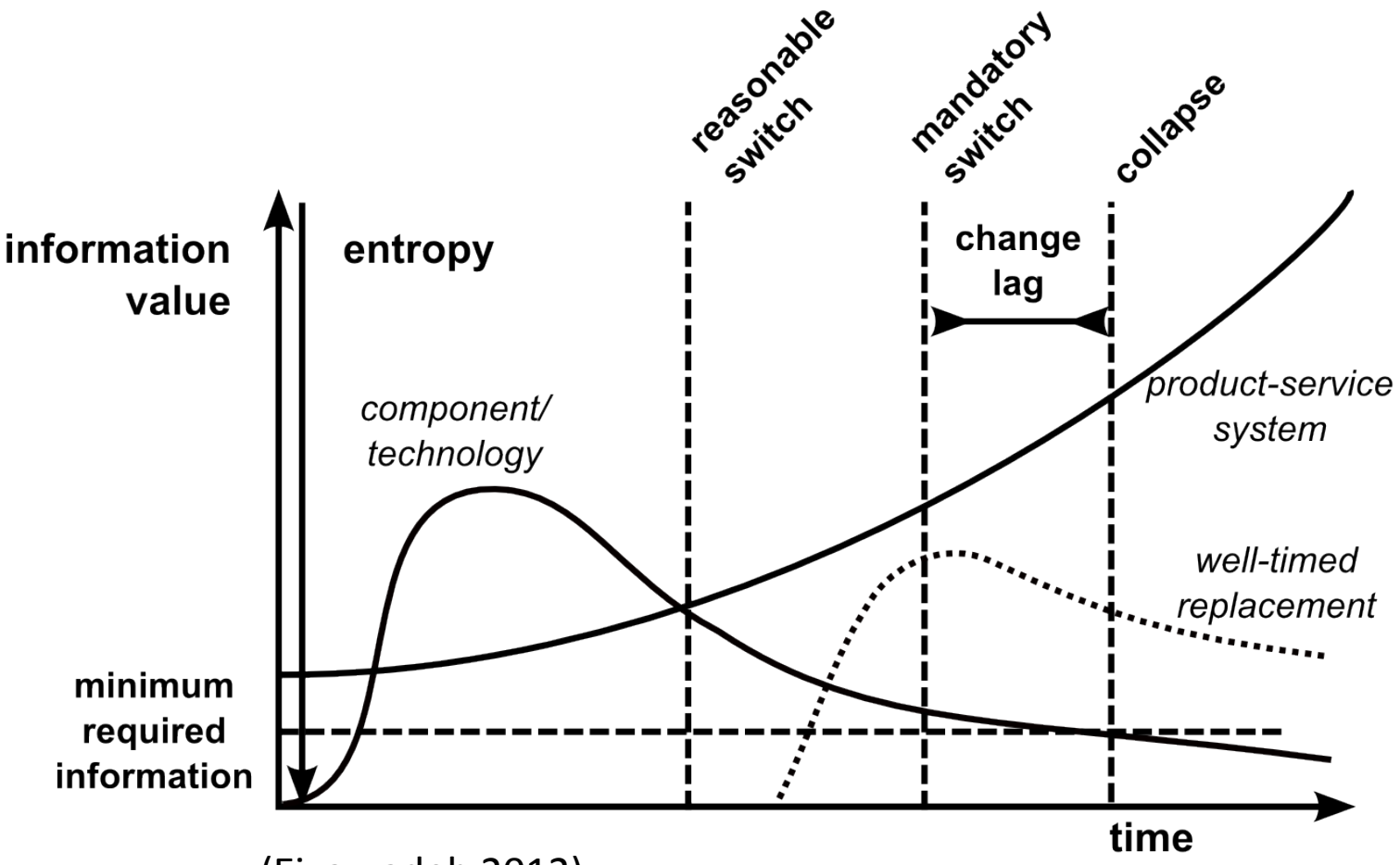
- Educating staff **vs.** recruiting smart members?
- Game consoles (learned/optimized about vector processing) **vs.** Powerful PC (higher CPU cycles)
- In the body immune system: number of Naive B-Cells **vs.** number of Memory B-Cells
- In evolution, the mutation rate (investing in possible future fitting genes **vs.** current stable genes)

#R Learning for Future vs. On-Time Processing Balance

- Expands the **latitude** with the same amount of resources



#P Information Ecdysis



(Eivazzadeh 2012)

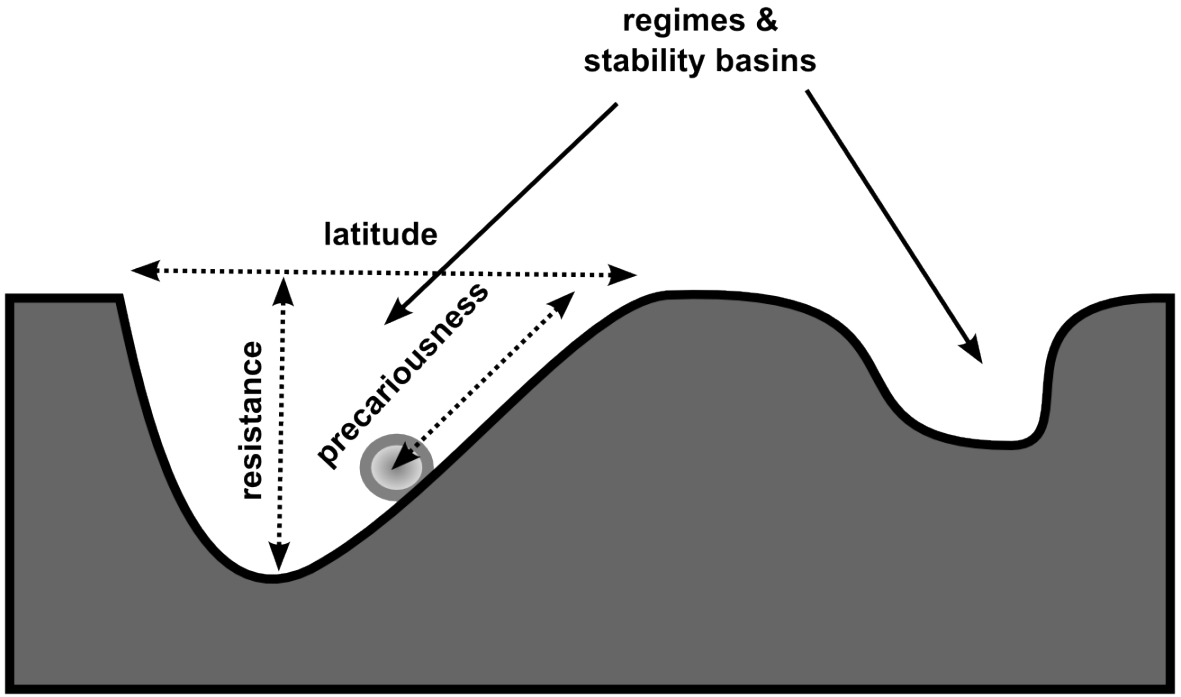
#P Information Ecdysis

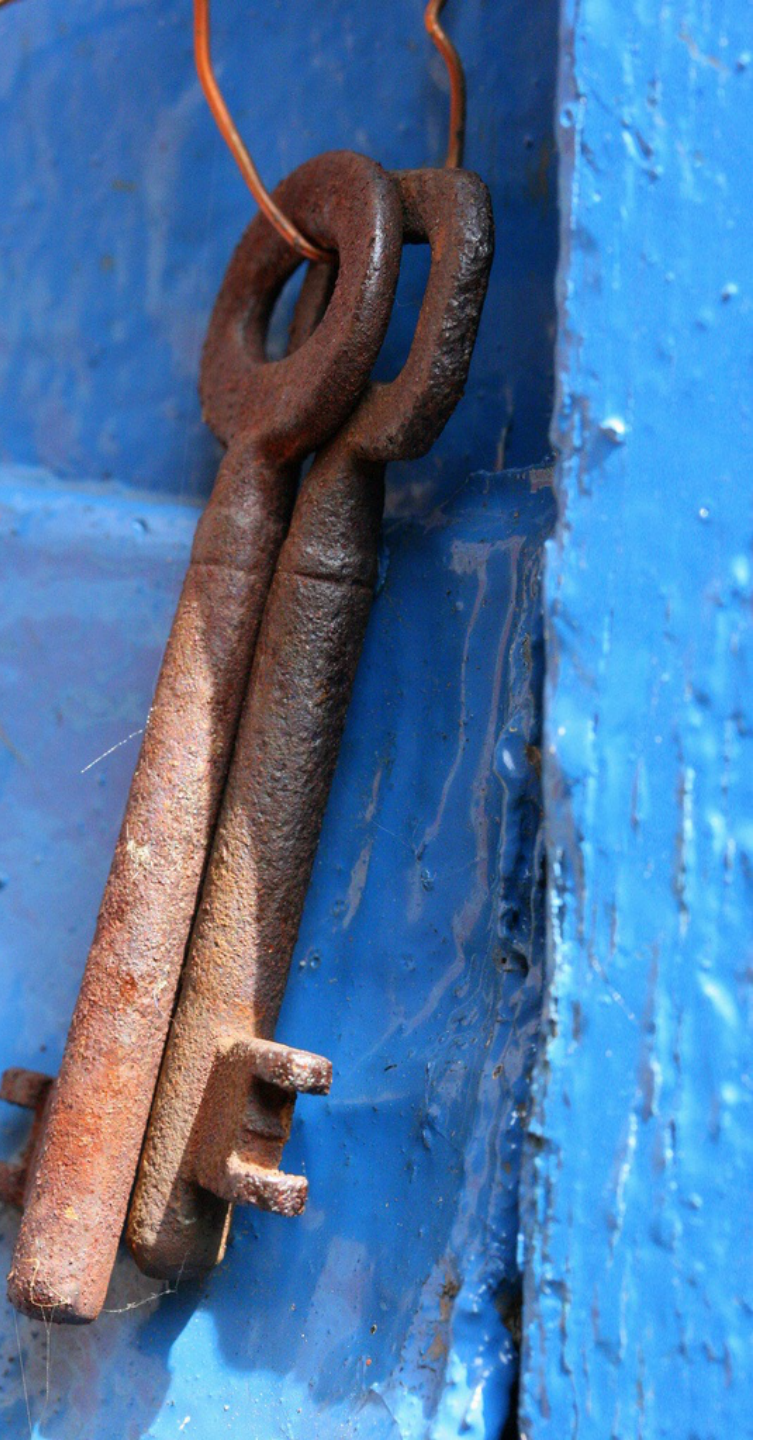
- Replace the people, replace the technology before any of them leave the designer team.
- NASA problem (till 2012): dependency on 8086 CPUs (first used 1981)
- Y2K (year 2000 crisis) problem led to problems from wrong embryo abortions to nuclear risks in Fukushima (first counter-measures initiated 1996)
- Y chromosome (which is in danger of extinction) has been replaced by other sex determination systems in some rodents.

Wikipedia, John Richfield

#R Information Ecdysis

- Decreases the **precariousness** of the system





Luc de Leeuw on Flickr

#P Smart Redundancy and Diversity

Harder to Fail by Creating **More Entropy** against the Adverse Environment!

- Smart redundancy avoids duplication of consumption (e.g. by focusing on important parts)
- Diversity adds more entropy than simple redundancy

Kerekes Jainos Csongor on Flickr

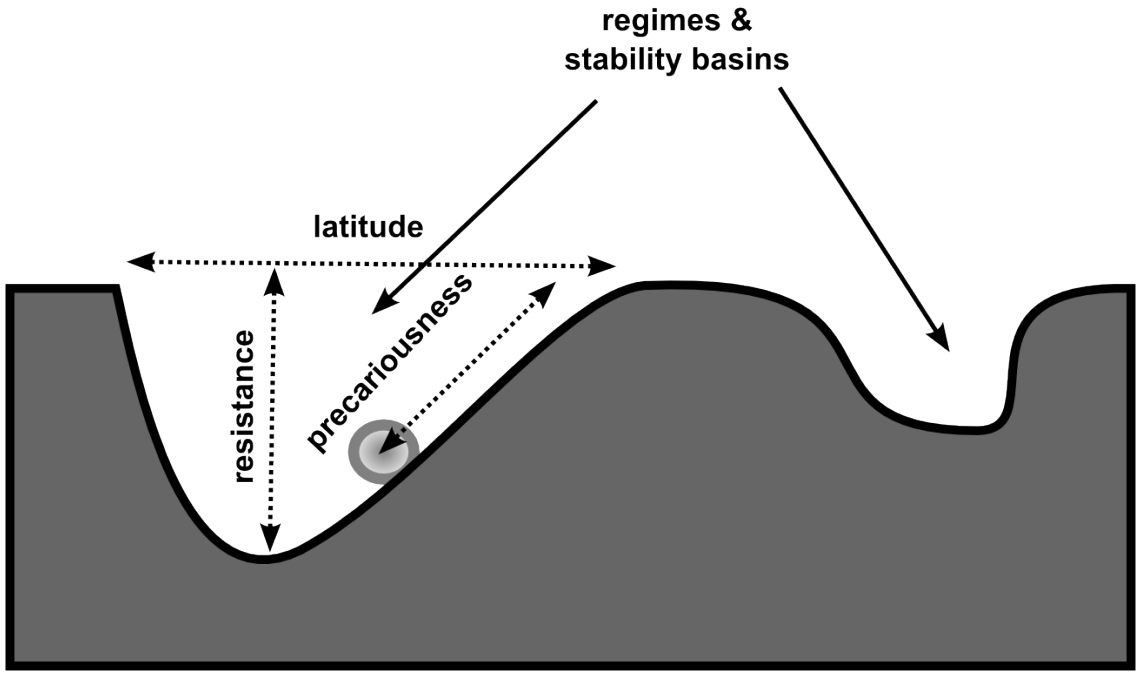
#P Regeneration

Smart Redundancy Along the Time

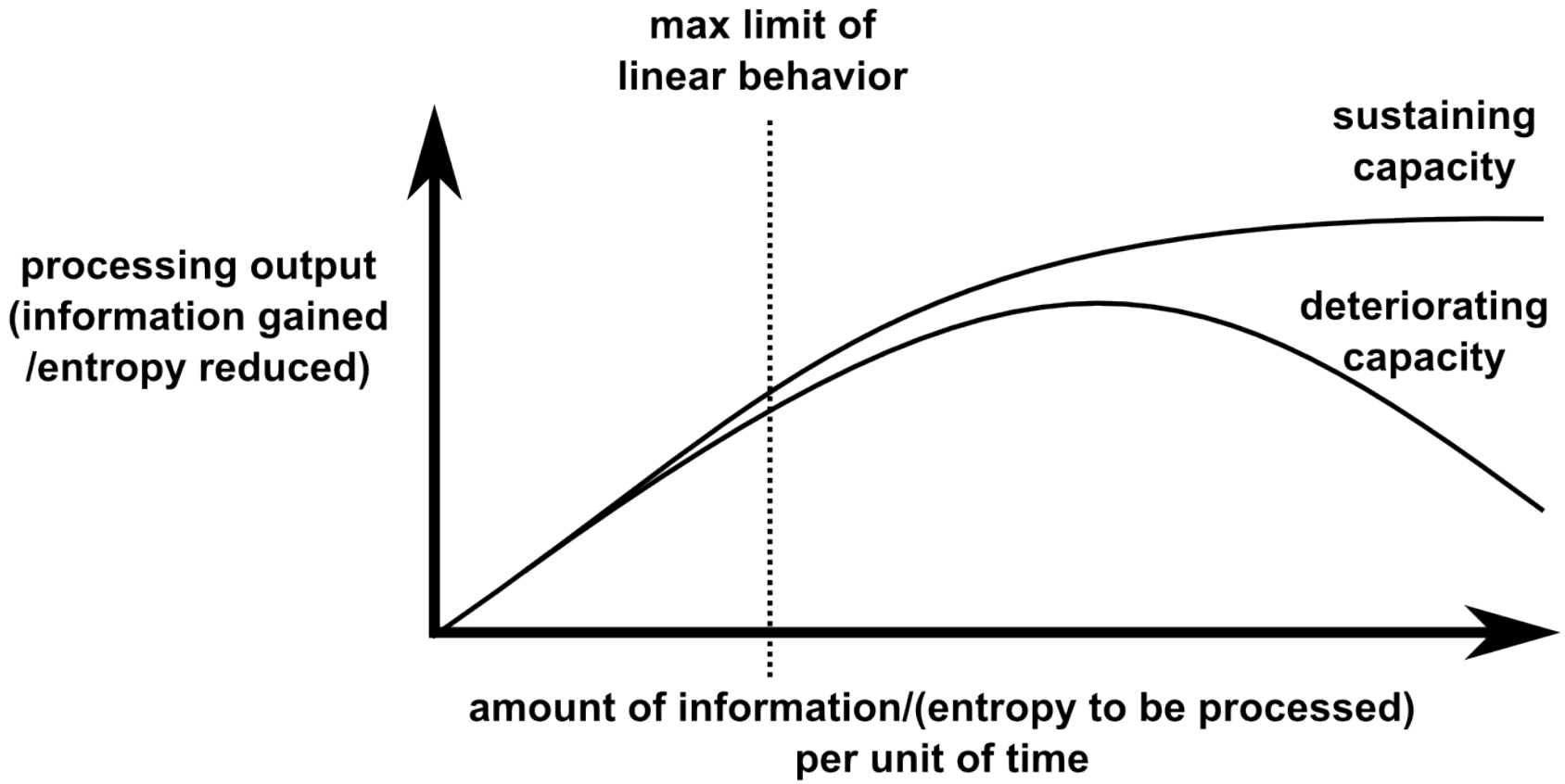
- Evolutionary regeneration is actually a diversity across the time

#R Smart Redundancy, Diversity and Regeneration

- Decrease **precariousness** by pushing back to the bottom of the basin



#P Processing Saturation Avoidance



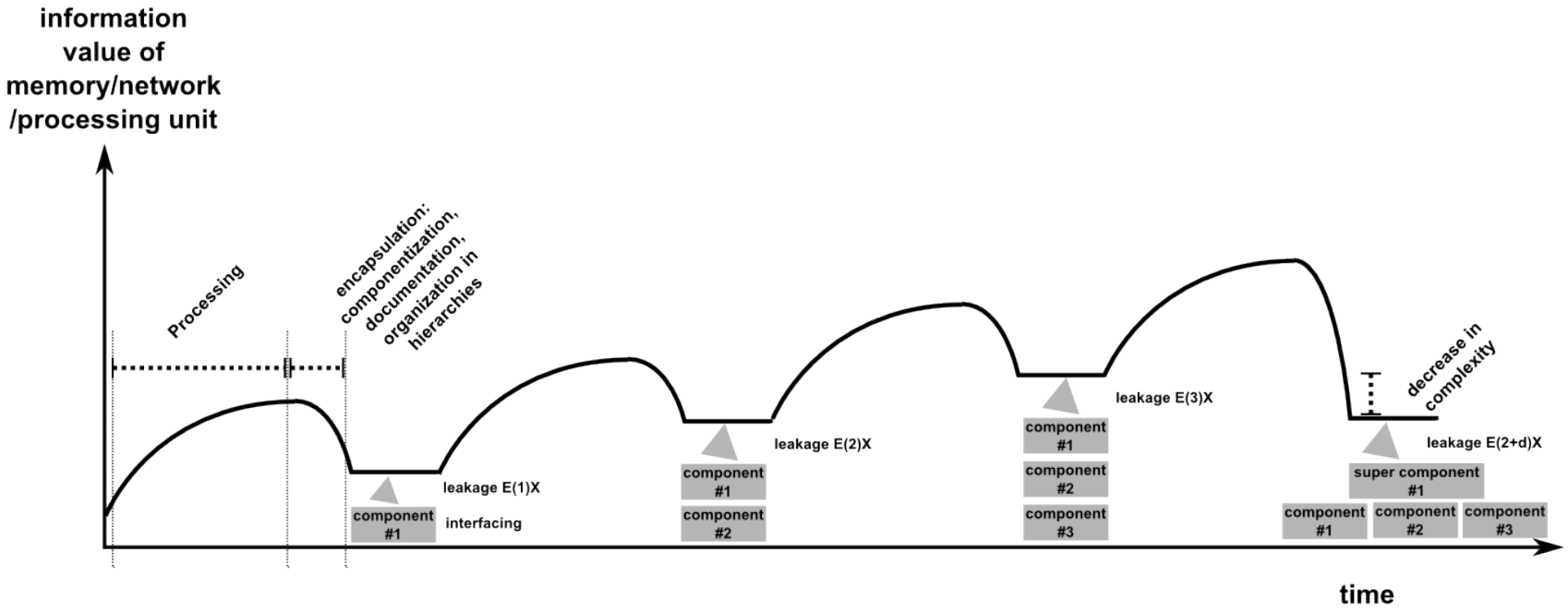
(Eivazzadeh 2012)

#P Refactoring

Same functionality, with less information, through restructuring

#P Encapsulation and Componentization

Hiding Information in Black-Boxes and Hierarchies



(Eivazzadeh 2012)

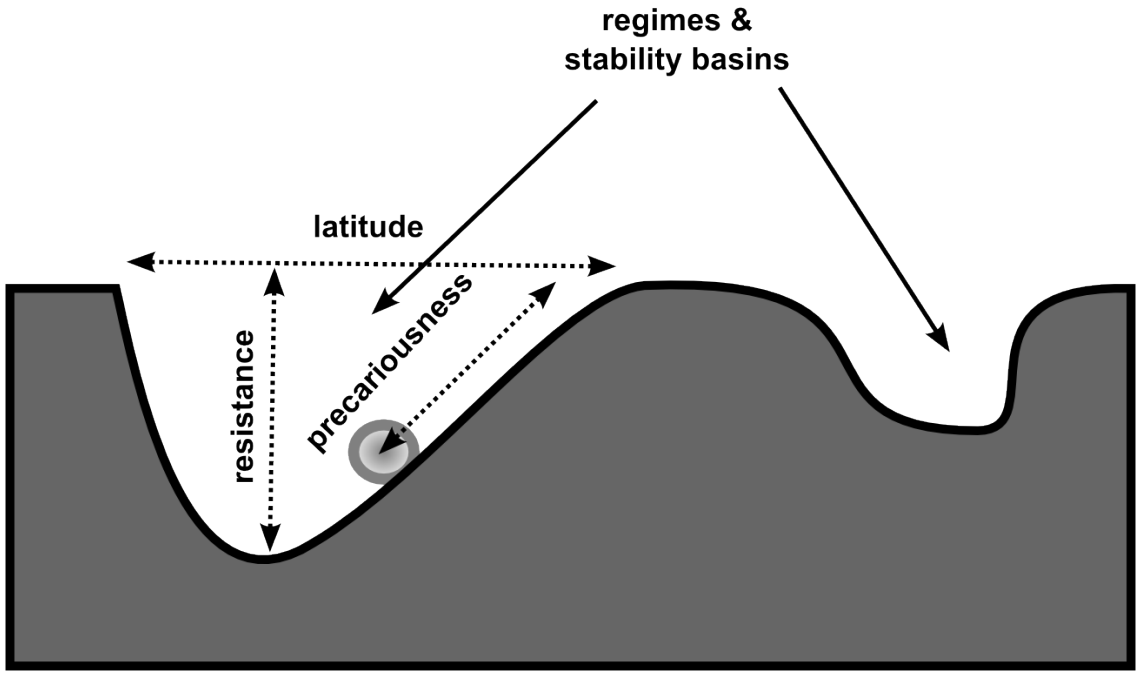
Yazd, Iran, Hercegnovi on Flickr

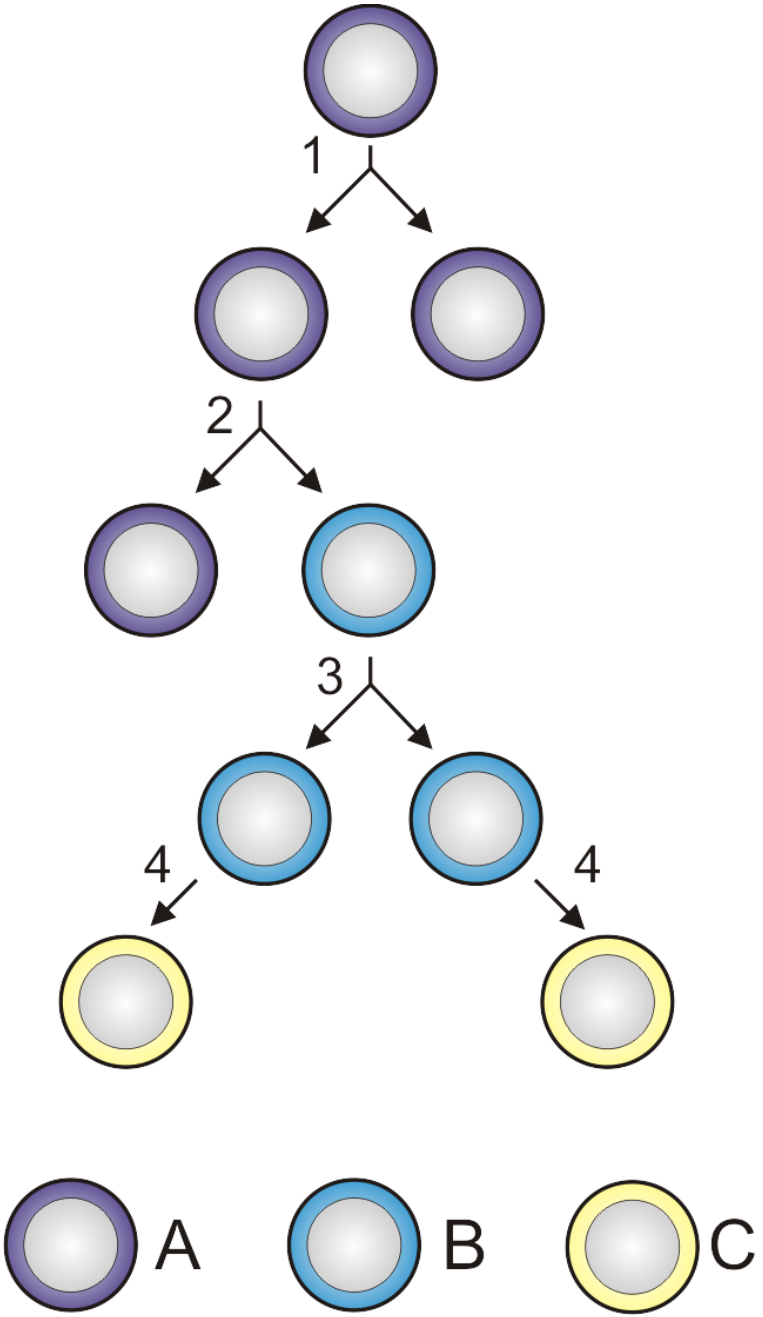
#P Recursion

- A type of componentization where the higher subsystems/components in hierarchy are very similar in structure and behavior with lower ones
- A kind of simpler and more compact encapsulation (less information intensive)
- Similar impacts on resiliency as componentization

#R Refactoring, Encapsulation, Componentization, and Recursion

- Decrease **precariousness** by pushing back to the bottom of the basin





Wikipedia, Public Domain

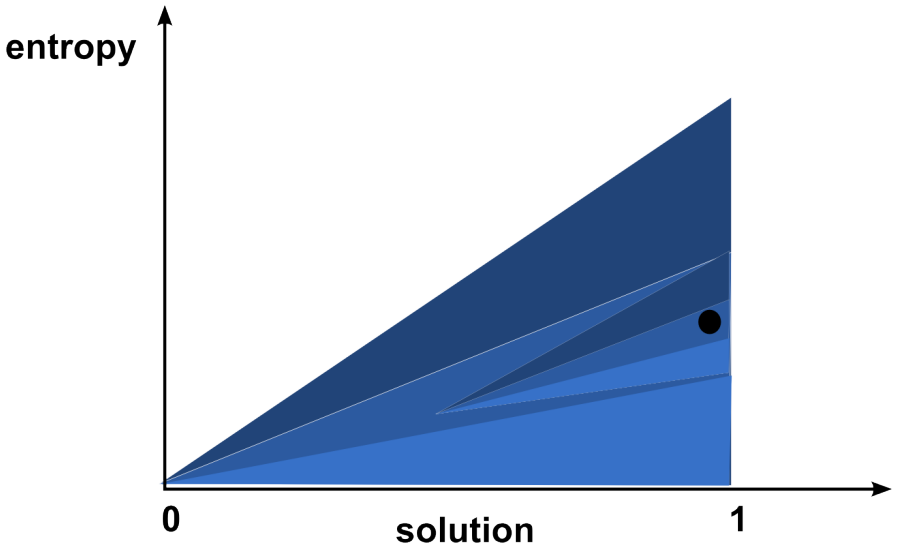
#P Stem-Cell

Information Traffic Congestion Shaping along the Time

- Begins from small teams of versatile members, ends in large teams of experts
- Can be applied in service teams
- Is already being used in agile software development methods

#P Cyclic Divergence and Convergence (CDC)

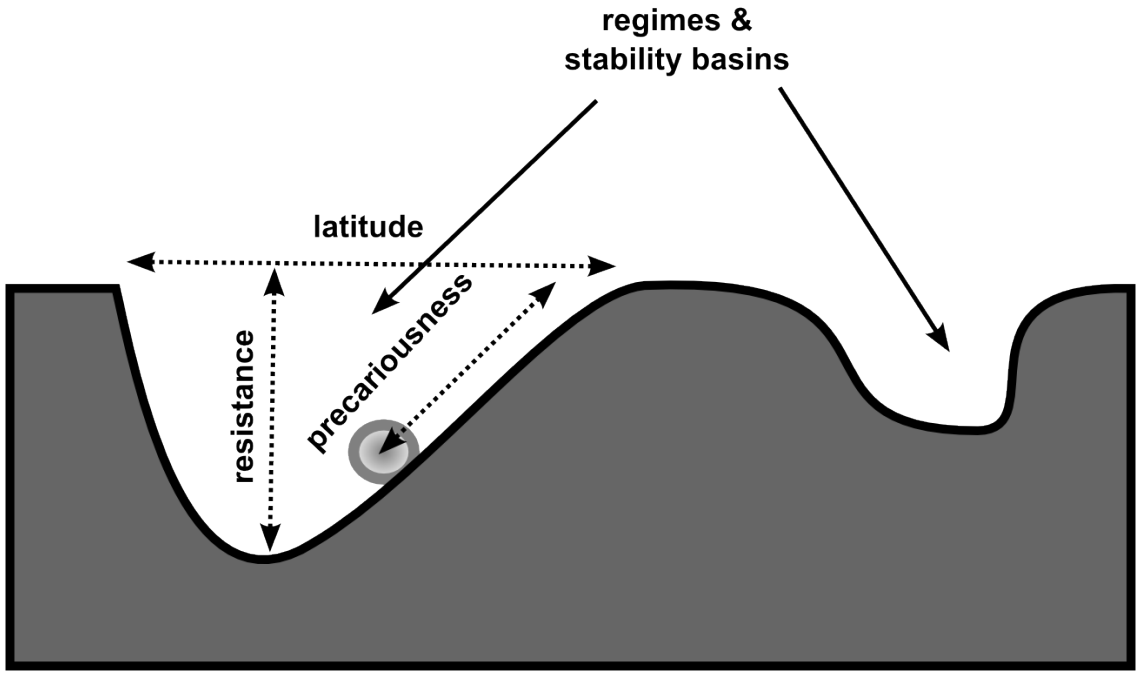
Divide and Conquer the Entropy
Information Traffic Congestion Shaping along the Time



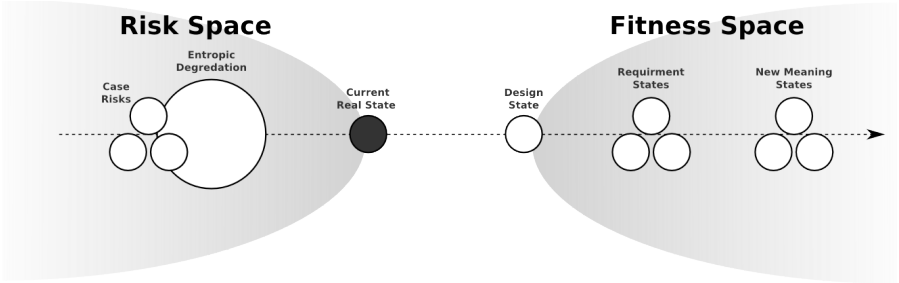
- Making the ultimate solution entropy space smaller
- Already being used (At least in extreme PSS innovation course!)

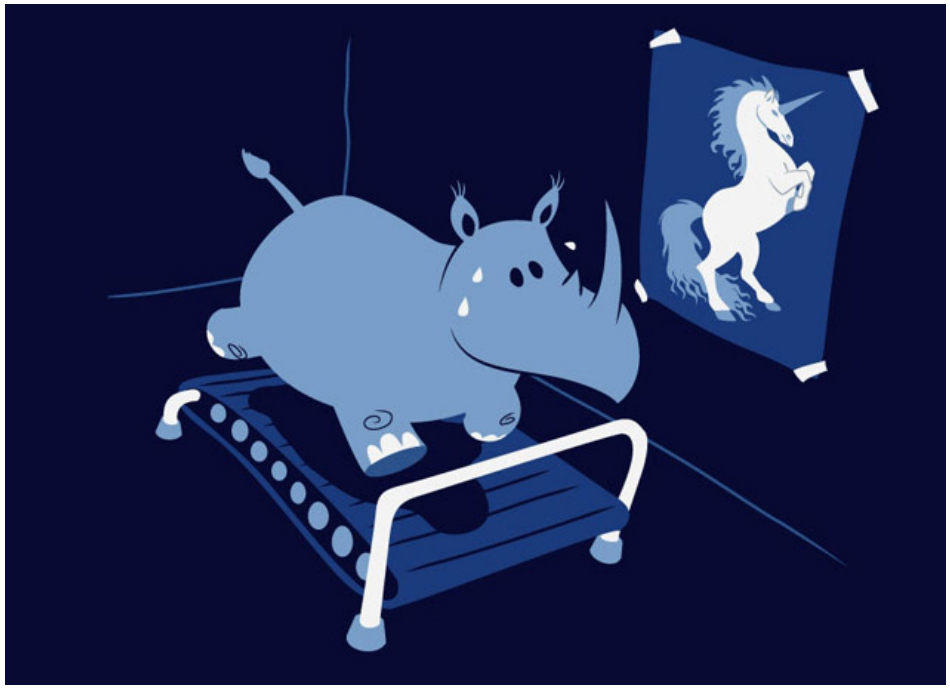
#R Stem Cell and CDC

- Decrease **resistance** by resisting overflows of information/entropy



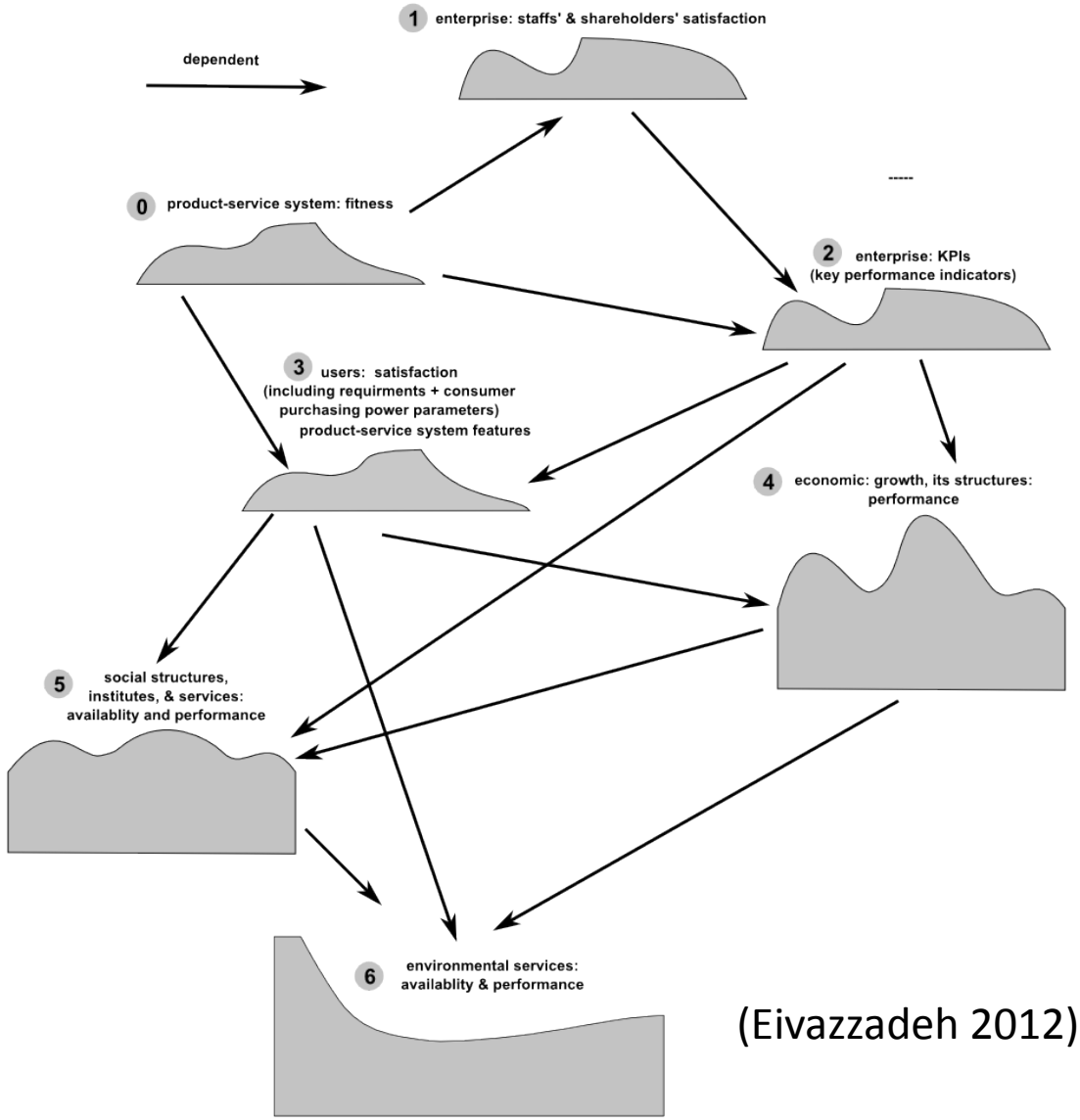
#P Multi-Level Fitness





Running Rhino, Credit: Allan Faustino, http://www.threadless.com/product/1000/Runnin_Rhino

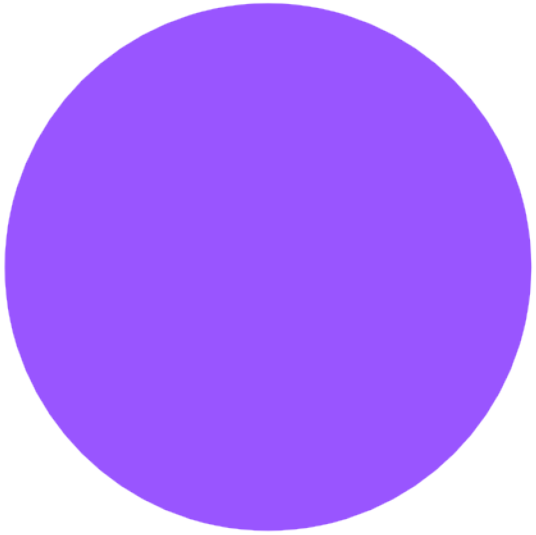
#P Multi-Level Fitness



(Eivazzadeh 2012)

#P Multi-Level Fitness

Shaping Entropy Space Considering Interrelation of Elements, their Importance, and Impact Propagation Delay



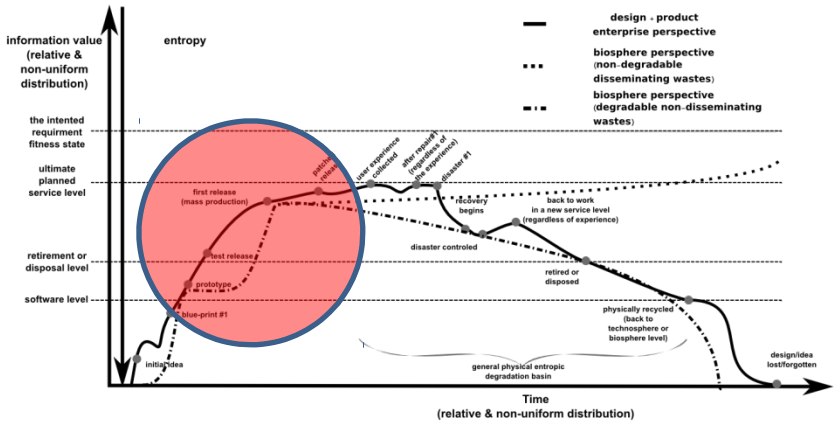
Simple Entropy Space

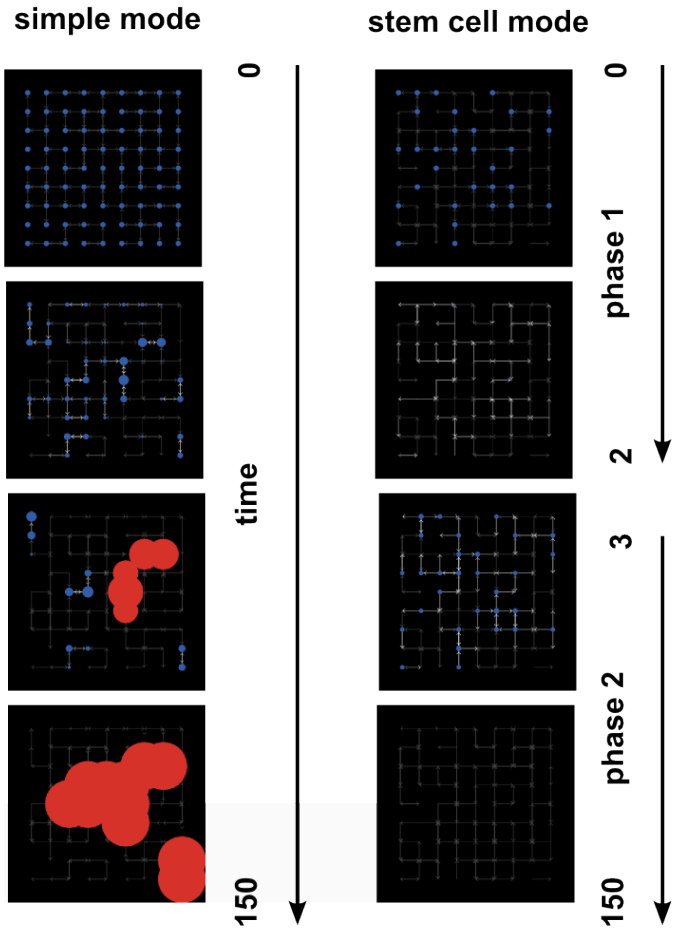
Weighted Entropy Space

Applications

#A Stem Cell

Simulation of Application in a Product-Service System





(Eivazzadeh 2012)

(Created by Netlogo 5)

… the sky's dome,
let us crack,
and come with a ***new design*** *…*

Hafez (1325-1389)