# Complexity, Antifragility, and Autonomic Networking

## Some Initial Thoughts

David Meyer
dmm@1-4-5.net

# Agenda

- What Flashed Together in My Mind
- Back to the Future: Historical Perspective/Framing
- What Is Complexity?
- What is Architecture?
- What Is Antifragility?
- How Might Autonomics Be Related?
- A Few Theses
- More Questions Than Answers

# What Flashed Together in My Mind

- Conjecture 1
  - The Autonomic Network is an RYF complex system who's organization (structure) is "designed" to create robustness
    - i.e., the *self-** properties
  - Corollary: The "autonomic control loop" can be studied as an instance of a set of layered robust control feedback loops that can be modeled as a RYF system

- Conjecture 2
  - An "Antifragile System" is RYF-Complex

- (C1 & C2) → Conjecture 3
  - The Autonomic Network is Antifragile
    - Or could be?

- So Obvious Questions:
  - Is there a unified theory underlying RYF, Antifragility, and Autonomics?
  - Does/would such a theory shed light on how to design/implement an autonomic network?

# Briefly, What Is Antifragility?

- Antifragility **is not the opposite** of fragility
  - ***Robustness*** is the opposite of fragility
  - Antifragile systems ***improve*** as a result of [perturbation]

- Metaphors
  - **Fragile**: *Sword of Damocles*
    - Upper bound: No damage
    - Lower bound: Completely destroyed
    - ***The cumulative effect of small perturbations is smaller than the single effect of a large perturbation***
  - **Robust**: *Phoenix*
    - Upper bound == lower bound == no damage
  - **Antifragile**: *Hydra*
    - Lower bound: Robust
    - Upper bound: Becomes better as a result of perturbations (within bounds)

- More detail on this later but just to give you a flavor

# Back to the Future
# Historical Perspective/Framing

- The use of the term ***organized complexity*** can be traced to Warren Weaver, who contrasted three classes of problems facing science [0]:
  - Problems of *simplicity*
  - Problems of *disorganized complexity*
  - Problems of *organized complexity*

- Let's briefly take a look at each of these

[0] W. Weaver, "Science and complexity," American Scientist, vol. 36, pp. 536–544, 1948.
See also H. Simon, "The Architecture of Complexity", Proceedings of the American Philosophical Society, Vol. 106, No. 6 (Dec. 12, 1962), pp.467-482.

# Problems of Simplicity

- Weaver briefly mentions simple problems as those involving a small number of variables, and he attributes much progress in the physical sciences and technology advances to successful application of the scientific method to simple problems
  - Note that modern computation has made the number of variables less important, however the distinctions remain relevant

- *Updated definition*: A system is simple if it has ***simple questions*** (i.e., models, theorems, experiments, and computations) to which there are ***robust answers***

- **Simple questions** are those that can be posed using models that are readily manageable and easy to describe, that theorem statements are short, and that experiments are elegant, are easily described, and require minimal interpretation

- **Robust answers** are those theorems have simple counterexamples or short proofs, algorithmic scale, and simulations and experiments are reproducible with predictable results

- Classical examples include: the pendulum as a simple harmonic oscillator; simple RLC circuits; the interaction of two bodies via gravity; and simple Boolean logic circuits as implemented in much digital hardware

# Problems of Disorganized Complexity

- From Weaver: Rather than pursuing simplicity with a few variables, "*imaginative minds went to the other extreme,*" ***focusing on problems with asymptotically infinite dimensions and developing "powerful techniques of probability theory and of statistical mechanics to deal with what may be called problems of disorganized complexity ."***

- Canonical example: In billiards, classical dynamics accurately predicts the trajectories of a small number of balls on a table, and expert players can robustly control these trajectories by keeping them relatively simple.
  - As the number of interacting balls increases, robust predictions become intractable, either computationally or for players
  - However, as the size of the table and the number of balls become very large, specific problems involving ensemble average properties actually become easier and more robust, and statistical methods apply, e.g., statistical mechanics

- In essence, what Weaver called disorganized complexity was ultimately a way to extend the "simple" to large ensembles and, in his thinking, was not really about complexity at all [AldersonDoyle2010]

- Chaos, criticality, scale free networks all fall into this class

# Problems of Organized Complexity

- Weaver used the term “disorganized” to emphasize that “the methods of statistical mechanics are valid only when the [billiard] balls are distributed, in their positions and motions, in a helter-skelter disorganized, way. i.e., randomly

- For example, the statistical methods would not apply if someone were to arrange the balls” and their movements in some highly organized manner.

- While Weaver acknowledges the prevalence of disorganized complexity in many important systems, he notes the importance of an intermediate class that “does not depend primarily on the fact that the number of variables is moderate. The really important characteristic of the problems of this middle region, which science has yet little explored or conquered, lies in the fact that ***these problems, as contrasted with disorganized situations with which statistics can cope, show the essential feature of organization*** . In fact, one can refer to this group of problems as those of organized complexity .”

- Organized complexity was formalized by Norbert Wiener (1961) in his development of cybernetics as a common theoretical core to the integrated study of technological and biological systems

- The rest of this deck focuses on Organized Complexity

# Contrasting Views

TABLE II
CONTRASTING VIEWS OF NSCN AND ORGANIZED COMPLEXITY

| | (a) New Sciences of Complex Networks | (b) Organized Complexity |
|---|---|---|
| **Primitives** | random ensembles | structured networks, architecture, design, evolution |
| **Function** | none | domain-specific performance and robustness |
| **Architecture** | graph topology, connectivity | protocols, layering, constraints that deconstrain |
| **Components** | largely homogeneous | heterogeneous, diverse |
| **"Not random"** | random but skewed, clustered | highly organized, structured, far from random |
| **Tuning** | *minimally tuned* via an order parameter | *highly tuned* via protocols |
| **Power laws** | "signatures" of criticality, SFN | "more normal than Normal" |
| **Uncertainty** | *minimal* | *large* (in environment and components) |
| **Robustness** | to random rewiring | to common perturbations, targeted attacks |
| **Fragility** | to common perturbations, initial conditions, attack | to random rewirings and rare perturbations |
| **RYF** | gratuitous | tradeoffs, unintended consequences |
| **Examples** | Citations, bacterial lineages, city sizes | Internet, biology, technology |

Table courtesy [AldersonDoyle2011]

# So What is *Complexity*?

"In our view, however, complexity is most succinctly discussed in terms of functionality and its robustness. Specifically, ***we argue that complexity in highly organized systems arises primarily from design strategies intended to create robustness to uncertainty in their environments and component parts.***" [AldersonDoyle2010]

# What is Complexity, cont…

- The observation is that protocols in such systems organize highly structured and complex modular hierarchies to achieve robustness, but also create fragilities to rare or ignored perturbations
  - Fat tailed distributions: low probability events can have massive consequences
  - → **Protocols are far more important to complexity than are modules**

- The evolution of protocols can lead to a robustness/complexity/fragility spiral where complexity added for robustness also adds new fragilities, which in turn leads to new and thus spiraling complexities

- The most powerful and also dangerous protocols involve feedback control, which also has the most mathematical and thus least widely understood theoretical foundations

- Finally, all of this complexity is largely hidden and deliberately creates the illusion of superficially simple systems, encouraging development of appealing and accessible but completely wrong explanations and theories (bummer)…cf edge of chaos, criticality, scale-free networks

# What Is Robust and What Isn't

- ***Definition***: A *[property*] of a *[system]* is **robust** if it is *[invariant]* with respect to a *[set of perturbations]*

- **Fragility** is the opposite of robustness (i.e., *–[invariant]*)

- A system can have a *property* that is *robust* to one set of perturbations and yet *fragile* for a *different property* and/or perturbation → the system is **Robust Yet Fragile (RYF)**

- Example: A possible ***RYF tradeoff*** is that a system with high efficiency (i.e., using minimal system resources) might be unreliable (i.e., fragile to component failure) or hard to evolve

http://www.istar.upenn.edu/osw/white paper/John Doyle White Paper.pdf

# Casting System Properties as Types of of Robustness

- **Reliability** is robustness to component failures
- **Efficiency** is robustness to resource scarcity
- **Scalability** is robustness to changes to the size and complexity of the system as a whole
- **Modularity** is robustness to structure component rearrangements
- **Evolvability** is robustness of lineages to changes on long time scales
- Question: How do these relate to the autonomic *self-** properties?

# Properties of RYF Systems

- **Narrow waists**
  - ATP metabolism, Internet Protocol
- **Massively distributed**
  - Metabolic pathways, BGP
  - Note OF/SDN
- **Highly layered**
  - Many immune system control loops
  - HTTP over TCP over IP over Ethernet over …
  - Note OF/SDN
- **Hidden robust control loops**
  - Cellular growth regulation, packet loss
- **Complexity-robustness spirals**
- Can be **arbitrarily fragile** to to derived properties
  - e.g., mean vs. variability
- All of this means that organized complexity is also about the management of functional robustness and the isolation of fragility
- Point here: highly evolved systems, ranging from advanced technologies to biology, exhibit RYF features, and understanding RYF tradeoffs lies at the heart of the design challenges for network-centric infrastructures

# Aside: Crutchfield on Layering and Modularity

- As they evolve, systems become more sophisticated, that is, structurally more complex → structural and behavioral correlation accumulates between components and across time
- A key and subtle step occurs when the structural relationships between the components, specifically their dynamical interaction, leads to a spontaneous architectural reorganization as new levels of pattern emerge.
  - That is, not only are individual components structurally complex and interconnected "horizontally" and in the process also become "vertically" nested
- The new patterns represent an increased level of abstraction in the system and reflect increased correlation (more structure) at a new level of organization
- The naive assumption of a system being composed of "modules"—in particular, that the modules are structurally or dynamically independent— fails. When correlation spontaneously emerges, the original components no longer need be "modules". They interface in new ways within the system and can give rise to new, unanticipated behaviors and functions that cross the system.
  - Moreover, these new functions can themselves become commandeered by other parts of the system.
- *Layer violations are an inherent part of organized complexity!*
  - Hence TCP/IP

See Crutchfield, J., "The Hidden Fragility of Complex Systems", Essay for Cultures of Change (Barcelona, December 2009)

# So…Complexity and Robustness

- Complex ***phenotype***: RYF
  - Think structure and function
- Complex ***genotype***: Internally complicated
  - Think control systems
- Applies to biological as well as technological systems
  - Pre-technology: simple tools
  - Primitive technologies use simple strategies to build fragile machines from precision parts.
  - Advanced technologies use complicated architectures to create robust systems from sloppy components…
  - … but are also vulnerable to cascading failures…

# RYF phenotype

- ***Robust*** to large variations in environment and component parts
  - reliable, insensitive, resilient, evolvable, simple, scalable, verifiable, …
- ***Fragile*** often catastrophically so, to cascading failures events
  - sensitive, brittle,…
- Cascading failures can be initiated by small perturbations
  - Cryptic mutations, viruses and other infectious agents, exotic species, …
- There is a ***tradeoff*** between
  - ideal or nominal performance (no uncertainty)
  - robust performance (with uncertainty)
    - Hence Robust rather than Optimal control systems
- Greater *pheno-complexity* → more extreme RYF

# RYF phenotype

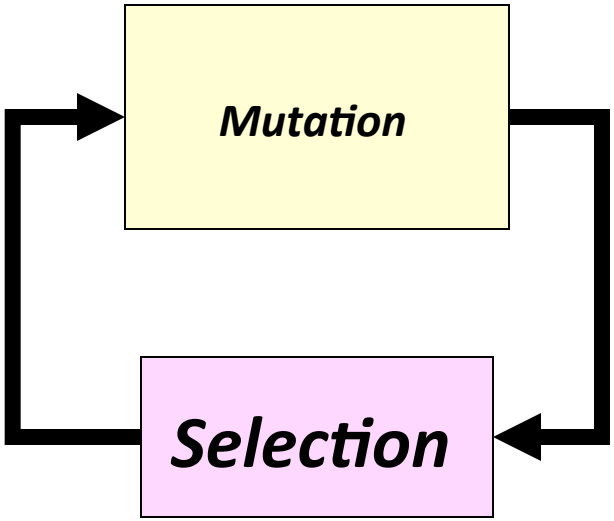
- In many complex systems, the size of cascading failure events are often unrelated to the size of the initiating perturbations
  - i.e., non-linear
  - What about Crowcroft, J., "Internet Failures – An Emergent Sea of Complex Systems and Critical Design Errors?", http://www.cl.cam.ac.uk/~jac22/out/bcs.pdf
- Fragility is interesting when it does not arise because of large perturbations, but rather when it creates catastrophic responses to small variations

# Complicated genotype

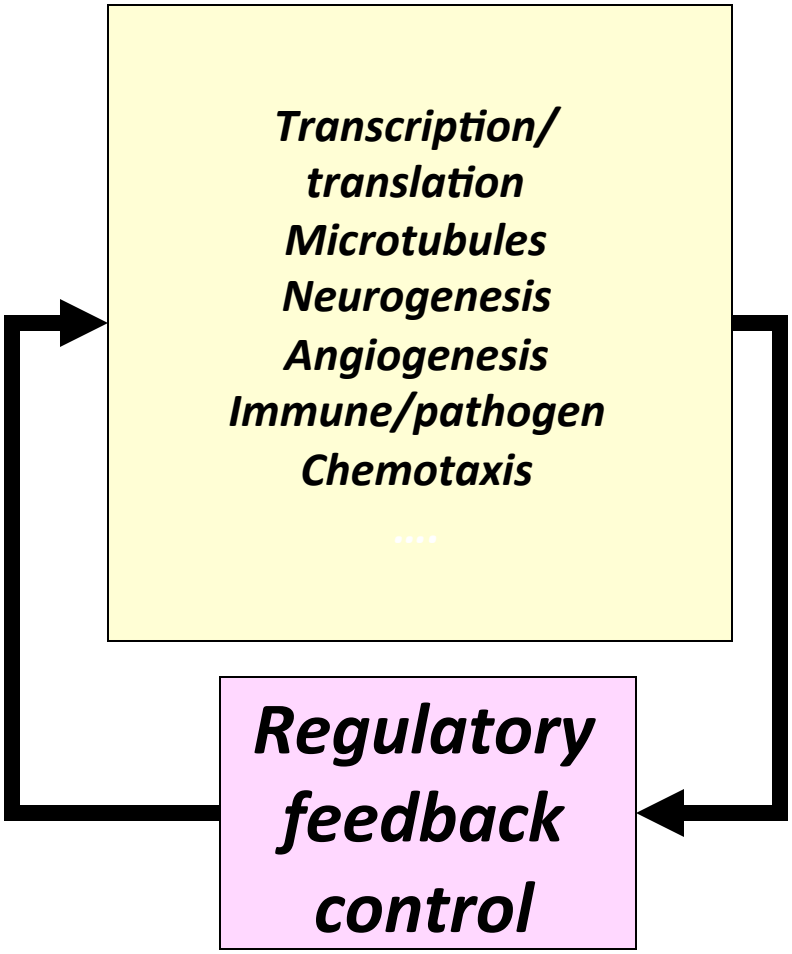
- Robustness is achieved by building barriers to cascading failures
- This often requires complicated internal structure, hierarchies, self-dissimilarity, layers of feedback, signaling, regulation, computation, protocols, ...
- Greater "geno-complexity" = more parts, more structure
- Molecular biology is about biological simplicity, what are the parts and how do they interact.
- If the complexity phenotypes and genotypes are linked, then robustness is the key to biological complexity
  - What about autonomic systems?
- BTW, "nominal functioning" may tell little in an RYF system

# An apparent paradox

Component behavior seems to be ***gratuitously*** uncertain, yet the systems have robust performance




Darwinian evolution uses selection on random mutations to create complexity.



# Aside: What's the Point of All of This?

- Understand the underlying design principles of these systems
  - to gain better understanding of their dynamics
  - provide design guidance to future systems designers
- If the Internet topology is the answer, what was the question?
  - If the TCIP/IP control loop was the question, …
- If the Krebs/Citric Acid Cycle is the answer, what was the question?
- But for us: ***Is Antifragility a form of RYF-Complexity and by transitive closure the Autonomic Network "could be" Antifragile***
  - And does any of this help us build and understand an Autonomic Network?

# Aside: Control Theory?

- **Optimal control** for additive noise (on average) can be arbitrarily fragile to other uncertainties
  - Led to the development of Robust Control Theory
  - Keep this in mind for later discussions of non-linearity (cumulative effects)

- **Robust control** is risk-sensitive, optimizing worst case (rather than average or risk-neutral) performance to a variety of disturbances and perturbations

- Robust control theory formalized and extended best practice in control engineering and coincided with a massive expansion into active control in robots, airplanes, automobiles, smart weapons, internets, …

- Robust control is generally hidden
  - For example, packet loss, skidding (anti-lock brakes),

# Aside: Control Theory?

- Control theory makes strong predictions about how robust circuits must necessarily be implemented largely independent of the device technology
  - Examples of robust control circuits: TCP/IP, autonomic control loops for *self-** functionality?
- Of special interest here: any natural parameterization of functional control is well-known to be large (high dimension), thin (even higher codimension), and nonconvex
  - Zhou K, Doyle JC, Glover K (1996) Robust and Optimal Control (Prentice Hall, Englewood Cliffs, NJ).
  - An amazing result we'll come back to later
- But for now, consider this simple example: Consider a 2D piece of paper with lengths that are large by some measure sitting in a 3D square box of comparable lengths
- The larger that these lengths are, the smaller that the fraction of volume that the paper will occupy in the box is.
  - i.e., if the area of the paper is X^2 then the fraction of the volume the paper represents decays as 1/X
- So the paper can be both large and thin as a fraction of the box volume
- Note that If the paper is bent or wrinkled, then it is also nonconvex within the box since most straight lines between two different points on the paper will not remain in the paper
- See Zhou K., Doyle JC, Glover K., "Robust and Optimal Control", Prentice Hall, Englewood Cliffs, NJ, 1996

# How Complexity Arises

- The specific structure highly organized systems is a consequence of specific ***constraints*** that are placed on their functionality and/or behavior
  - Importantly, these consequences are *largely independent* of the process by which this organization arises
  - i.e., whether by design or evolution

- Four kinds of constraints
  - component
  - system/environment
  - protocol
  - emergent

- Let's briefly look at each of these

# Component-Level Constraints

- The components that comprise any system are typically constrained in terms of what they can do
  - even separately
  - ultrareliability vs. uncertain reliability

- For example, much of mechanics, electrical circuits, chemical processes, etc., can be described in terms of relationships such as F = MA and V = IR
  - Of course, these constraints are often expressed as differential or algebraic equations
  - But also much more general

- The **uncertainty** of components often imposes constraints on a complex system that are as important as the nominal idealized component behavior

# System-Level Constraints

- What distinguishes biology and technology from other types of systems is that there are complex constraints on the system as a whole that **are not consequences of those on the components**, including functional requirements,

- These include
  - What the system needs to do
  - Environmental and operating requirements required to achieve robustness
  - Robustness to uncertainty and perturbations from the environment

- Is “intention” a system level constraint?

# Protocols

- Protocols allow communication with the external environment
- Protocols typically take the form of rules for the configuration and/or interaction of system components, may impose additional constraints on the overall system
  - Is intention a “protocol”?
- Although these additional constraints may reduce the number of possible system solutions, a “good” set of protocols minimally constrains these solutions so as to facilitate a focus on the feasible and robust solutions

# Emergent Constraints

- The interaction of the previously mentioned constraints can imply an additional set →
- "Emergent" constraints that are **nontrivial consequences** of the interaction between the system and component-level constraints, and possibly protocols
- Perhaps the most important emergent property of any set of constraints **is whether their intersection is (non)empty**, so theory and methods to determine this are central to engineering specification and design
- Emergence is also associated with ***unintended consequences*** for either good (an emergent benefit) or bad (an emergent fragility).

# What is *Architecture*?

- Architectures are fundamentally about handling or creating the various constraints described previously to facilitate "good" solutions among competing tradeoffs
  - Can be designed, evolved, or both

- No coherent theory of architecture (yet)

# Protocol Based Architectures (PBAs)

- Protocols are a basic part of organized complexity
- Gerhart and Kirchner [1] nicely capture the role of protocols in the biological domain with the phrases "constraints that deconstrain" and "facilitated variation."
  - They describe how constraints in the **form of universal shared protocols** provide a platform for diverse functionality and robustness by *facilitating* large but functional *variation* on which selection can act
- PBAs allow typical network behavior to be fine-tuned through elaborate but hidden control systems and thus appear boringly robust despite large internal and external perturbations
- For *PBAs,* the *protocols* (rules of interaction that persist) are more fundamental than the *modules* (which obey protocols and can change and diversify)

[1] Gerhart, J., and M. Kirschner, "The theory of facilitated variation", http://www.pnas.org/content/104/suppl.1/8582.full.pdf

# Protocols and the Internet

- The TCP/IP protocol suite enables adaptation and control
  - time scales vary from the sub-microsecond changes in physical media to the millisecond- to-second changes in traffic flow, to the daily fluctuations in user interactions, and to evolving hardware and application modules over years and decades
  - Hidden, robust controller

- *Layering* is perhaps the most important architectural feature
  - captures the idea that each layer in the protocol stack provides services to the layer above by utilizing and **hiding** the resources in the layers below

- The protocol stack is called an "hourglass" because a thin hidden "waist" ***of universally shared feedback control (TCP/IP)*** layers sits between a vast diversity of visible upper (application software) and lower layers.
  - Provides the performance and robustness of both the "horizontal" decentralized and asynchronous nature of control in TCP/IP as well as the "vertical" separation into the layers of the TCP/IP protocol stack from application down to the link layer

- Internet vs. biological systems
  - The upper Internet layers are all software
    - In biological systems all layers involve chemistry
  - The Internet imports electric power and hardware from external manufacturing processes
    - Bacterial cells, OTOH, are massively autocatalytic
    - cf "smart net"
  - That said, there are striking similarities at the architectural level

# RYF in PBAs

- Bacterial and Internet PBAs underlie both their robustness and fragility
  - PBAs allow typical network behavior to be fine-tuned **through elaborate but hidden control systems** and as such appear boringly robust despite large internal and external perturbations
- PBAs also facilitate evolution
  - from microbes to humans
  - from an academic research network to a global information infrastructure
- An important result of layering and control is that complexity and fragility remain ***largely hidden***, often revealed only by catastrophic failures and often only after the system has absorbed multiple insults without apparent consequences
  - Remember this for the discussion on non-linearity
  - Is the cumulative effect of the "multiple insults" related to RYF behavior?
- Basically, large structured rearrangements are tolerated by control systems that reallocate network resources, easily conferring robustness to outright failures of components and brute force attacks,
  - However, cost of this robustness is disastrous fragilities to small random or targeted changes that subtly violate core protocols
- The greatest fragility in PBAs is from parasites and predators, who hijack and consume universal and standardized interfaces and building blocks

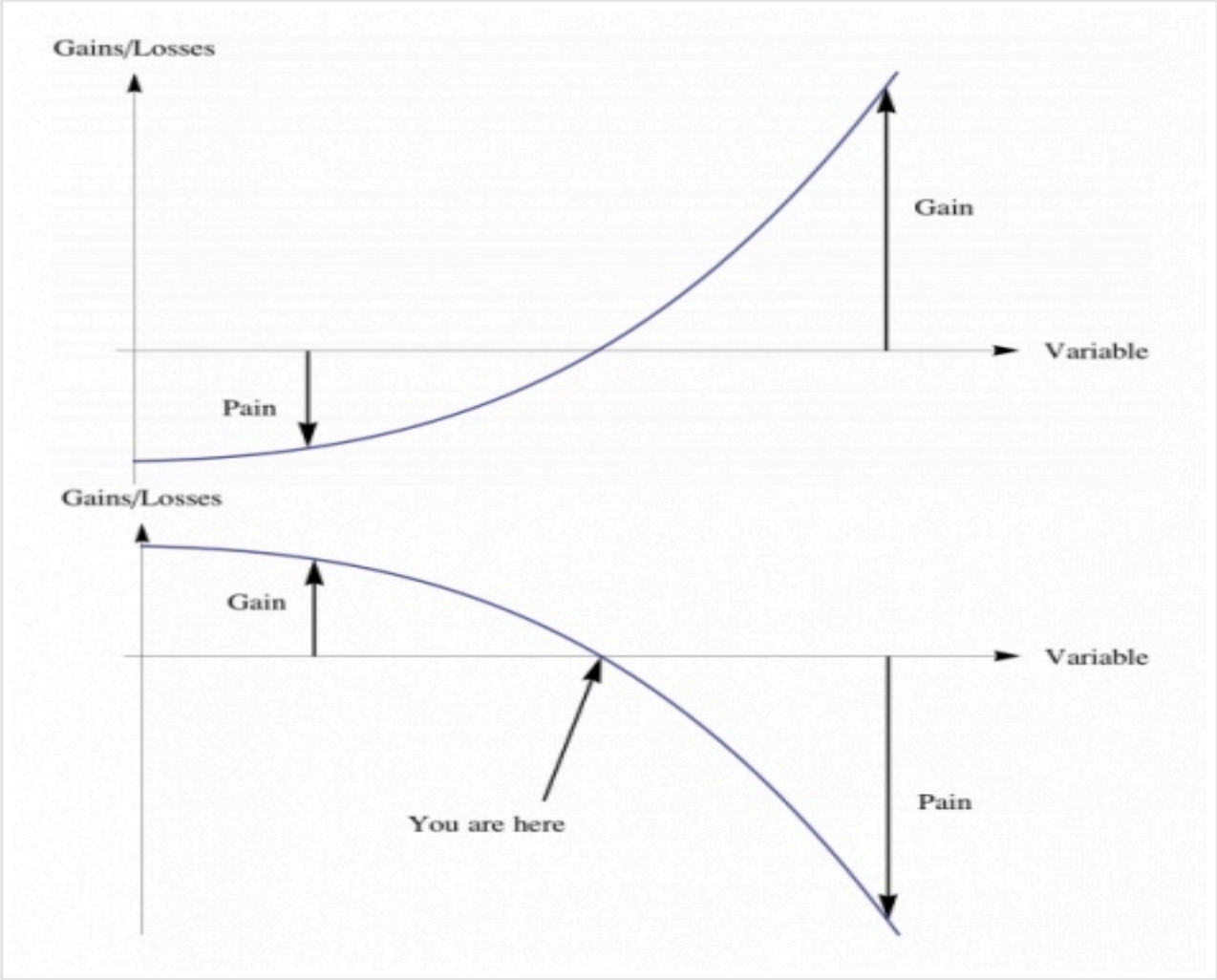
# What then is *Antifragility*?

- Concept from a different domain [Talab2011]
  - Motivated by Risk Engineering Theory and barrier option theory
  - Question: Do antifragile systems share common phenomenology with RYF systems?
    - i.e., is this really about organized complexity?

- Antifragility **is not the opposite** of fragility
  - Antifragility requires thin left-tail (exponential decline of probabilities) and local convexity, usually expressed as positive sensitivity to the dispersion parameter of the probability distribution
    - so-called "long vega" in options theory
    - more on this in subsequent slides

- Metaphors
  - **Fragile**: *Sword of Damocles*
    - Upper bound: No damage
    - Lower bound: Completely destroyed
    - ***The cumulative effect of small perturbations is smaller than the single effect of a large perturbation***
  - **Robust**: *Phoenix*
    - Upper bound == lower bound == no damage
  - **Antifragile**: *Hydra*
    - Lower bound: Robust
    - Upper bound: Becomes better as a result of perturbations (within bounds)

# Quick Review: Convex vs. Concave (aka No Pain No Gain)



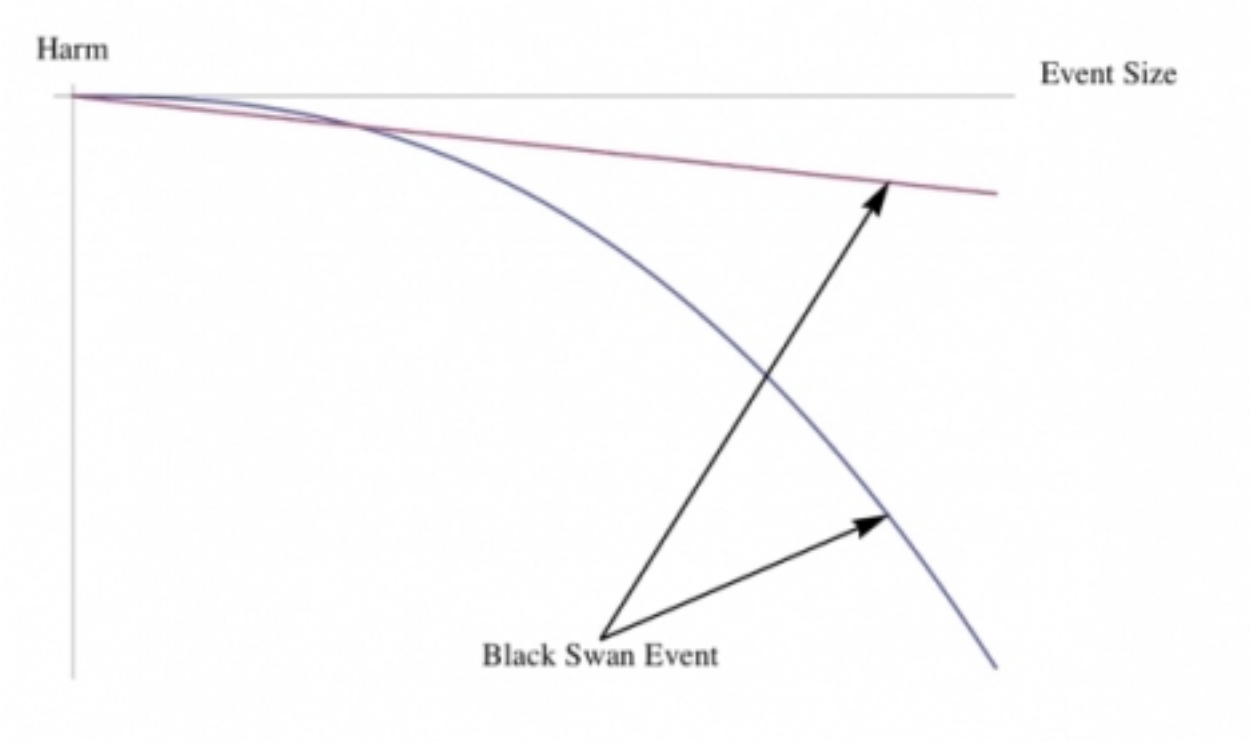
# What is Antifragility All About?

- Its really about ***the convexity of the response*** of a system's [property] to a [set of perturbations]
  - Non-linear response → system is either fragile or antifragile
    - BTW, do we have non-linear properties in the network?
    - Again, see Crowcroft, J., "Internet Failures – An Emergent Sea of Complex Systems and Critical Design Errors?", http://www.cl.cam.ac.uk/~jac22/out/bcs.pdf

- If response is **concave** (negative convexity), then you are fragile
  - For the fragile, the **cumulative effect** of small shocks is smaller than the single effect of a large shock
  - Notice that nature requires this property
    - Consider jumping off a 1 foot high block 30 times vs. jumping one time from 30 feet
  - Basically if the response weren't non-linear the cumulative effect of walking around would have long ago killed us

- If response is **convex**, then you are antifragile
  - For the antifragile, shocks bring more **benefits** (equivalently, less harm) as their intensity increases
  - as always, up to some bound

# Linear vs. Non-Linear Harm



# Negative Convexity Effects

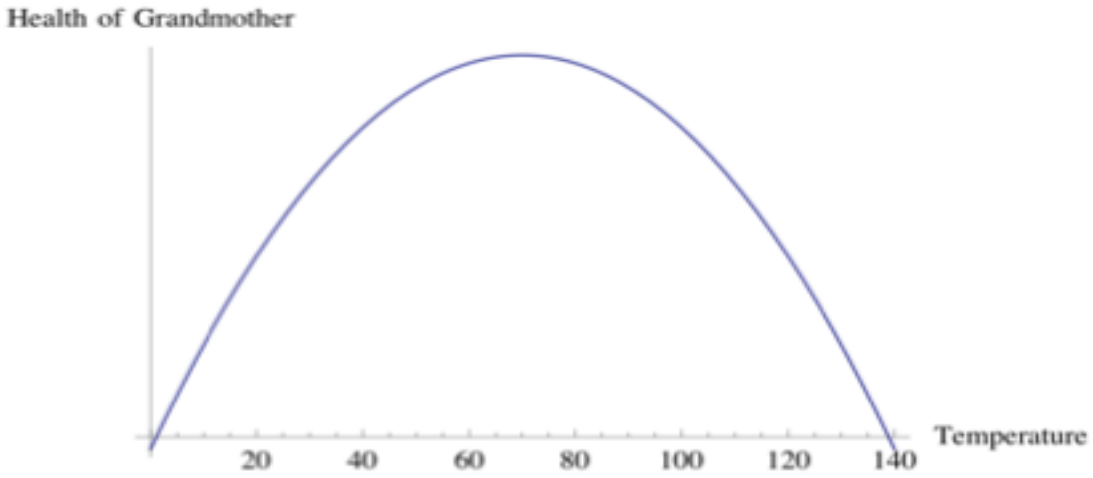



*Figure 12* ***Fragility****: Health as a function of temperature curves inward. A combination of 0 and 140 degrees (F) is worse for your grandmother's health than just 70 degrees. In fact almost* ***any*** *combination averaging 70 degrees is worse than just 70 degrees*. The graph shows concavity or negative convexity effects —curves inward.*

- Grandma's health is fragile to variation in temp (this graph), but not to the mean temp, within bounds
- Negative Convexity → the "fragility" in RYF systems?
  - Grandma robust to mean temp, fragile to variation (i.e., RYF?)
- Mean temp and variation → system-level constraints

# Another Example

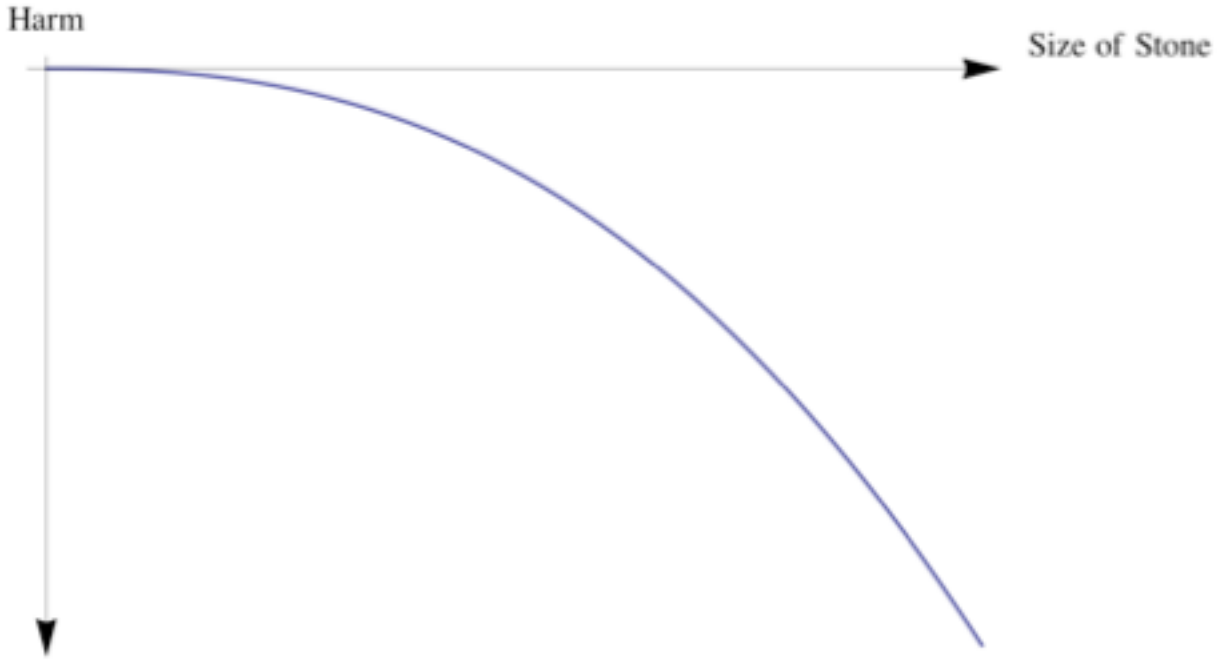



*Figure 7- The King and His Son. The harm from the size of the stone as a function of the size of the stone (up to a point). Every additional weight of the stone harms more than the previous one. You see nonlinearity (the harm curves inwards, with a steeper and steeper vertical slope).*

# Hormesis

(what doesn't kill you makes you stronger…*up to a point*)

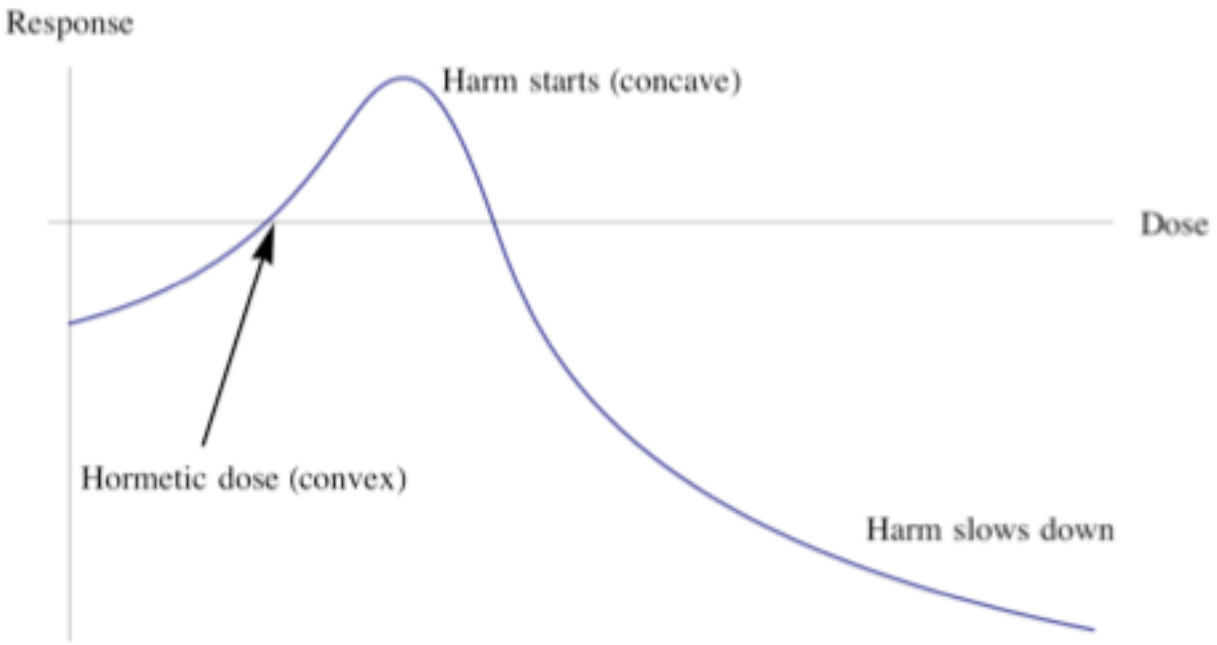



*Figure 17- Hormesis for an organism: we can see a stage of benefits as the dose increase (initially convex) slowing down into a phase of harm as we increase the dose (initially concave), then things flattening out at the level of maximum harm (beyond a certain point, the organism is dead so there is such a thing as a bounded and known worst case scenario in biology) – Note that (medical papers and textbooks make the mistake of having concave curve at the early stages, which would be mathematically impossible).*

# Antifragility can (must) be Local

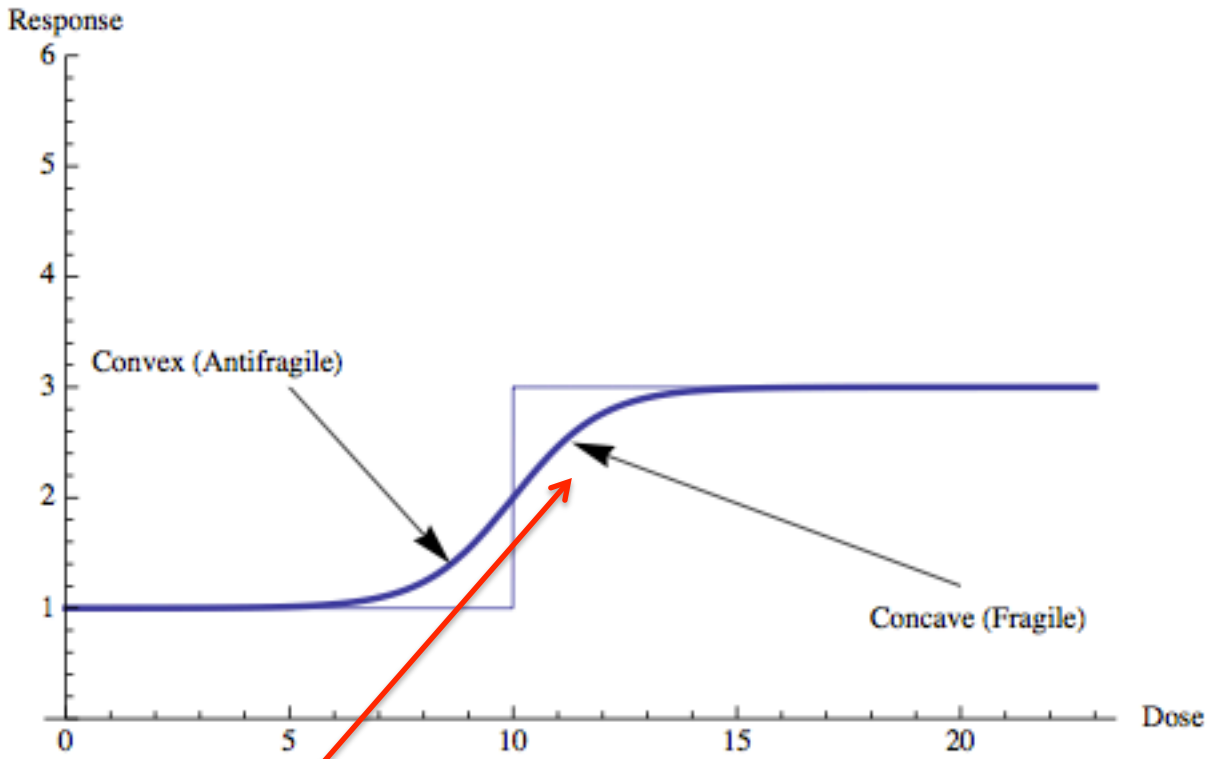



Figure 13- Dose response in biological systems (similar to digital payoffs). The mixture regularizes say, caloric consumption.

**“up to some point”**

# Another Definition of Fragility

- Fragility ⇔ Left Tail → Concavity
  - Negative convexity (see table on next slide)

- In this formulation Fragility equates the sensitivity of left tail shortfall (non conditioned by probability) to increase in disturbance over a certain threshold K

- Basically: Fragility → left tail, Antifragility → right tail

- Examples
  - A porcelain coffee cup subjected to random daily stressors from use
  - A tail distribution in the function of the arrival time of an aircraft
  - Hidden risks of famine to a population subjected to monoculture
  - Hidden tail exposures to budget deficits' nonlinearities to unemployment
  - Hidden tail exposure from dependence on a source of energy, etc.
    - so-called "squeezability argument"

# Types, Conditions, and Tails

| Type | Condition | Left Tail (loss domain) | Right Tail (gains domain) | Nonlinear Payoff Function $y = f(x)$ "derivative", where $x$ is a random variable | Derivatives Equivalent (Taleb, 1997) | Effect of Jensen's Inequality on Missed Nonlinearities | Effect of Fat tails in Distribution of primitive $x$ |
|---|---|---|---|---|---|---|---|
| **Type 1** | **Fragile (type 1)** | Fat | Thin | Concave | Short gamma | Lower expectation | Worsens |
| **Type 2** | **Fragile (type 2)** | Fat (regular or absorbing barrier) | Fat | Mixed concave left, convex right (fence) | Long up – gamma, short down – gamma | Lower expectation in case of absorbing barrier | Worsens if absorbing barrier, neutral otherwise |
| **Type 3** | **Robust** | Thin | Thin | Mixed convex left, concave right (digital, sigmoid) | Short down – gamma, long up – gamma | Invariant | Invariant |
| **Type 4** | **Antifragile** | Thin | Fat (Thicker than left) | Convex | Long gamma | Raises expectation (particularly in Type 4 *b* where trigger barriers cause ratchet – like properties) | Improves |

# Autonomics: Problem Statement

"The essence of the autonomic-computing vision is the human nervous system which is capable to maintain the body in the appropriate equilibrium by interpreting thousands of parameters and accordingly adapt body functions in a very complex manner. " [0]

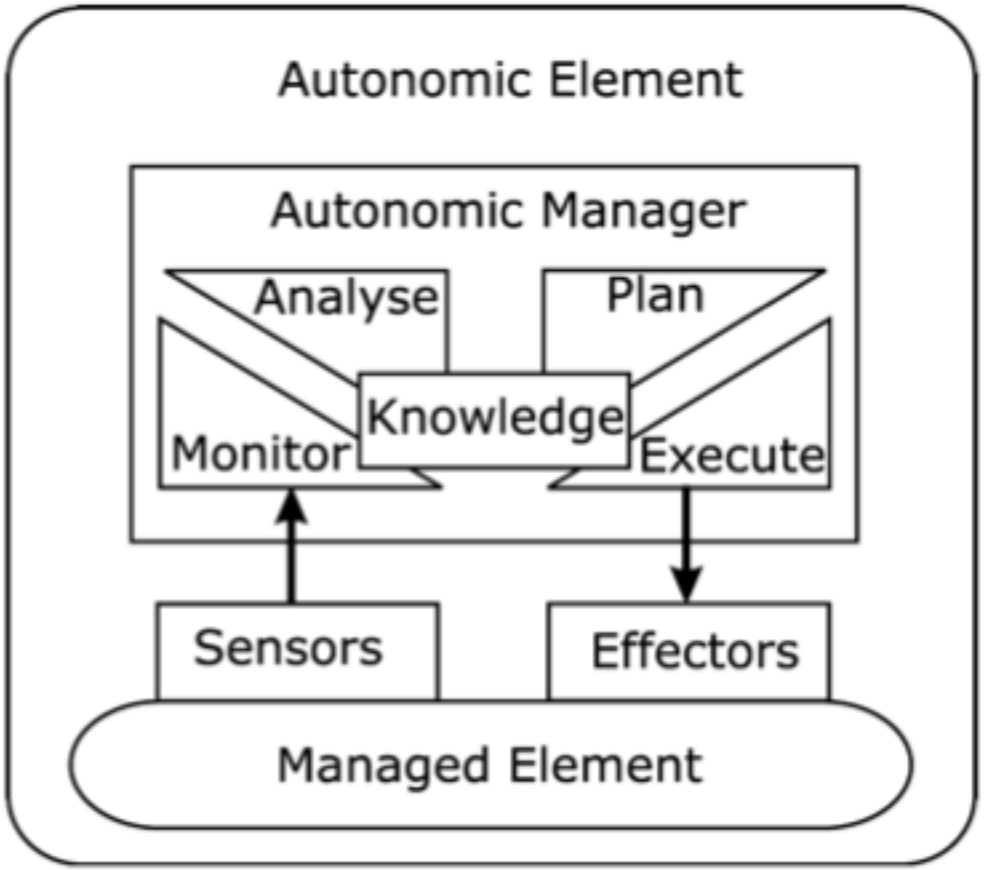
→ Autonomics is "bioinspired"

Another more functional definition: ***Autonomics is concerned with the deployment of technology specifically to manage and optimize the functioning of other technology on an ongoing basis***. In many ways, autonomic design seeks to generalize the control-theoretic view of control by enabling more flexible and adaptive functions in the underlying system (though I believe this might encompass a somewhat dated view of the state of the art in control theory [dmm]).

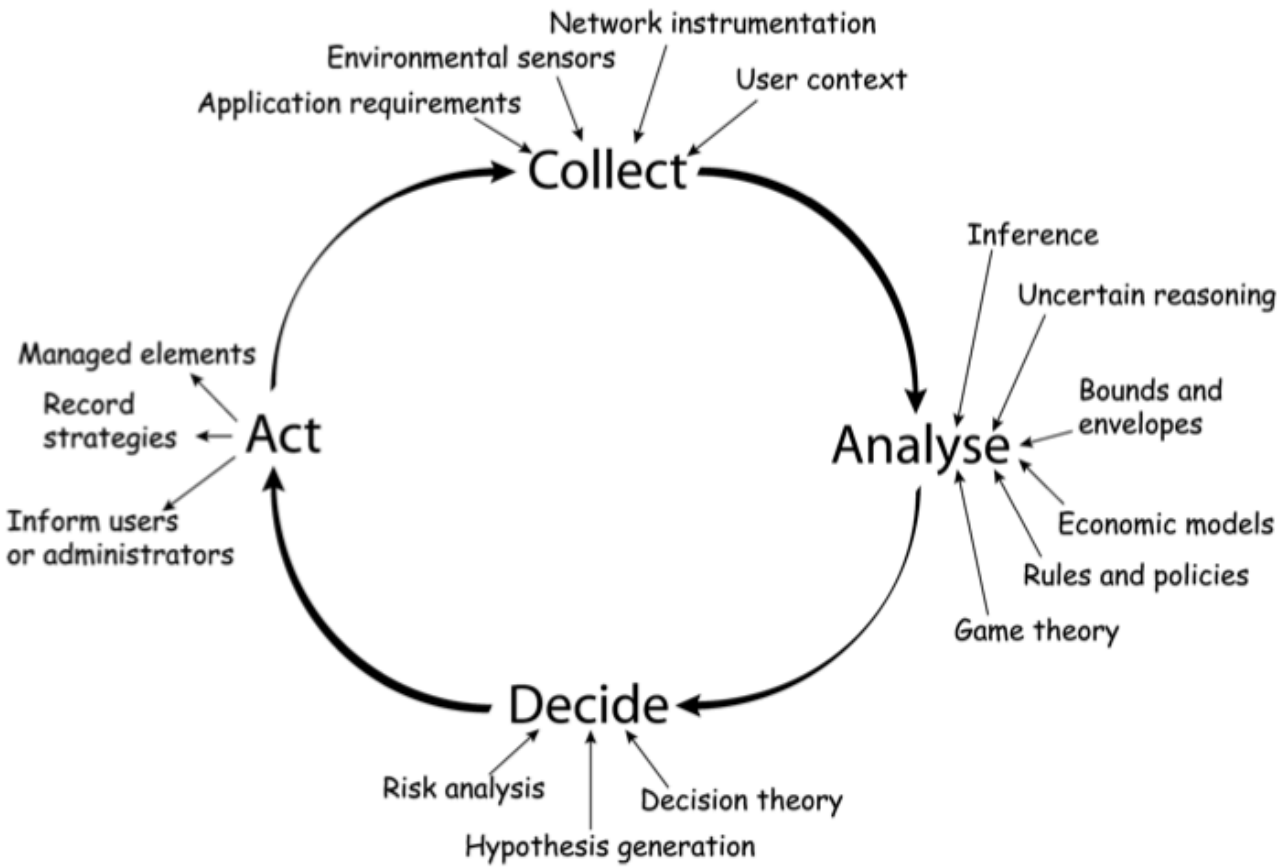
[0] Agoulmine, N., et. al., "Challenges for Autonomic Network Management"

# Autonomic Control Loops: MAPE-K

[Monitor, Analyze, Plan, Execute, Knowledge [IBM 2003]]



# Autonomic Control Loop Cartoon



# Modern Control Systems

- Modern control systems are tpically implemented using digital hardware and software, and most computers are embedded in this way and thus, are ***permanently hidden***

- Examples are ubiquitous:
  - antilock brakes
  - automated collision avoidance
  - global positioning systems in cars
  - fly by wire aircraft.
  - …

- The internal mechanisms are again manifest largely in the ***rarity*** of crashes, losses, errors, and failures and in the ***catastrophic*** nature of rare crashes

- However, despite enormous progress, robots struggle to navigate the real world as effectively as rodents or even insects, and computers continue to fail in Turing tests, although in fascinating ways that reveal much about both humans and computers
  - Autonomics?

- → This enormous, hidden, cryptic complexity, driven by robustness, is both the greatest initial obstacle in using advanced information and control technologies as metaphors for biology and also ultimately, the key to important insights and theories

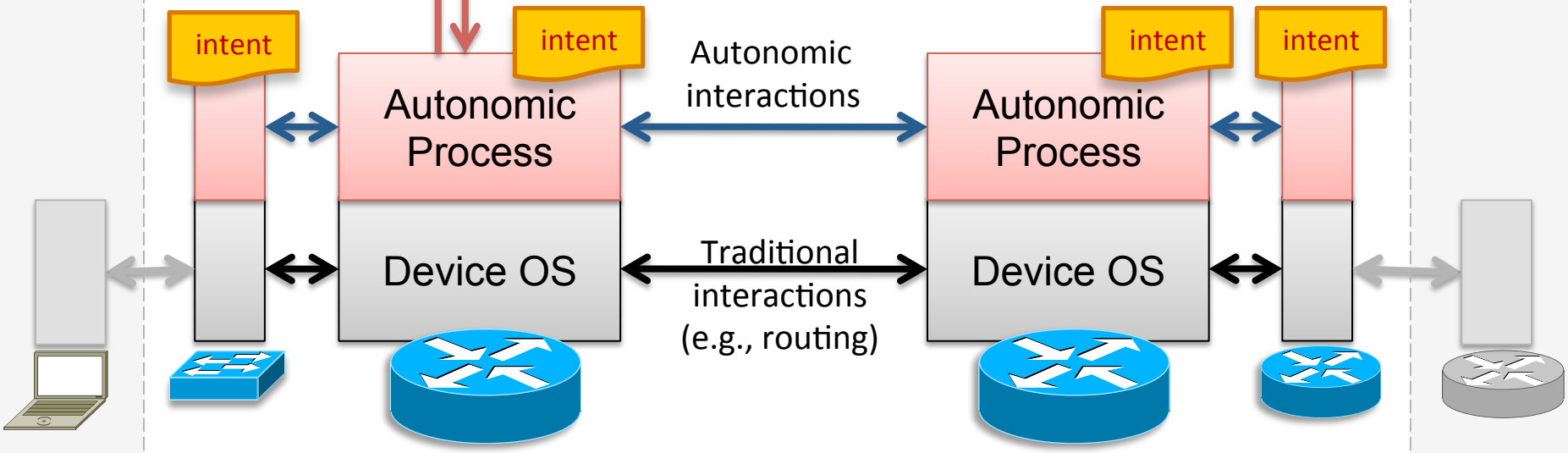
# The Autonomic Network


- Autonomic Interaction include Autonomic Adjacency Discovery and Capability Exchange → Network builts itself.
- Intent is described network wide and executed locally, together with local defaults and gleaned information from neighbours → Network wide abstraction

Simple Management tools


feedback

intent

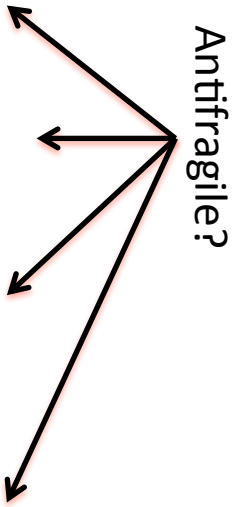



Slide courtesy Yves Hertoghs

# Autonomics and Antifragility

**Table 1. Four aspects of self-management as they are now and would be with autonomic computing.**

| Concept | Current computing | Autonomic computing |
|---|---|---|
| Self-configuration | Corporate data centers have multiple vendors and platforms. Installing, configuring, and integrating systems is time consuming and error prone. | Automated configuration of components and systems follows high-level policies. Rest of system adjusts automatically and seamlessly. |
| Self-optimization | Systems have hundreds of manually set, nonlinear tuning parameters, and their number increases with each release. | Components and systems continually seek opportunities to improve their own performance and efficiency. |
| Self-healing | Problem determination in large, complex systems can take a team of programmers weeks. | System automatically detects, diagnoses, and repairs localized software and hardware problems. |
| Self-protection | Detection of and recovery from attacks and cascading failures is manual. | System automatically defends against malicious attacks or cascading failures. It uses early warning to anticipate and prevent systemwide failures. |




Is some composition of *self-* antifragile?*

See Kephart, J., and D. Chess, "The Vision of Autonomic Computing "

# More Questions Than Answers (1)

- Is antifragility an instance of organized complexity?
  - Pretty clearly true
- Are antifragile systems RYF?
  - Is antifragility an extension of RYF theory or covered by RYF theory?
  - Conjecture: Yes, antifragility is RYF/HOT
- Can we ***engineer*** an Antifragile network?
  - Can we build anything antifragile?
    - Antifagility is "emergent" (in the RYF sense)
    - Fromm, J., "On Engineering and Emergence ", http://arxiv.org/pdf/nlin/0601002.pdf
  - Engineered Emergence [Anthony 2006]?
    - "The purposeful design of interaction protocols so that a predictable, desired outcome or set of outcomes are achieved at a higher level"
    - This is an approach to building systems that benefit from characteristics such as scale, robustness, and stability, but do not require precise knowledge of lower-level activity or configuration
  - Network is already RYF
    - The F in RYF is negatively convex (theorem; no proof but intuitively seems so)

# More Questions Than Answers (2)

- The idea that robust systems are ***large*** (high dimension) but ***thin*** (even higher co-dimension) and ***nonconvex*** in the space of all systems is a theme in RYF literature
  - Example: Consider the set of words in most languages, which is large but vanishingly thin as a fraction of all possible meaningless sequences of letters
    - There are 9! = 362,880 different permuted sequences from just the nine distinct letters *adeginorz*, roughly the total number of English words, but only *organized* is a word
  - Example: The set of functional parameter values of any circuit will also typically be large but vanishingly thin and nonconvex in the set of all possible (mostly nonfunctional) circuits
    - like autonomic control loops?

- How does this theory relate to the set of layered autonomic control loops?

# More Questions Than Answers (3)

- Autonomics seems to have antifragile and RYF properties
  - And "intent" is a sort of system-level and/or protocol constraint
  - → meant to structure the system to produce specific kinds of robustness
  - Is some composition of *self-* properties* antifragile?
  - What frameworks might be available to think about this?
    - RYF/HOT theory?

- So
  - Can we formalize Antifragility in terms of RYF theory?
  - Can we build a formal ***theory that frames autonomics as an RYF antifragile system***?
  - Can we create the "antfragile network"?
    - That is, antifragile with respect to which properties?
    - And how would this be parameterized/analyzed?
  - Application to sensor networks/internet of things….

# More Questions Than Answers (4)

- Is there literature/theory that relates the idea of autonomics to organized complexity?
  - literature seems surprisingly sparse
- In any event, how to reverse engineer a theory that is useful in implementing autonomic systems?
- BTW, isn't *sensor networking* just an instance of an autonomic network?
  - How about cloud?