Bio-techno Convergence and The Hidden Nature of Complexity

David Meyer
CTO and Chief Scientist, Brocade
Director, Advanced Technology Center, University of Oregon
Network Complexity Research Group
IETF 88
Vancouver, BC
dmm@{brocade.com,uoregon.edu,1-4-5.net,…}
http://www.1-4-5.net/~dmm/talks/ncrg88.pdf

Agenda

- Goals/Premise/Thesis
- Robustness, Fragility, and Complexity
- Bio/Techno Convergence
- Universal Architectural Principles
- A Few Conclusions and Q&A if we have time

Goal of this Talk

To open up our thinking about what the essential architectural features of our network are, how these features combine to provide robustness (and its dual, fragility), and how the universal architectural features we find in both technological and biological networks effect Internet robustness.

Premise

Biological and advanced technological systems are *robust* and *evolvable* even in the face of large changes in environment and system components, yet can simultaneously be extremely fragile to small perturbations.

This *Robust Yet Fragile* (RYF) complexity is found wherever we look. Remarkably, a key feature of such RYF-complex systems is that it is the same universal architectural features that makes these systems robust and evolvable also creates severe, even catastrophic fragilities to tail events.

Thesis

Advanced technological networks and biology exhibit striking similarity (*convergence*[1]) at "higher" levels of organization.

As such there is a lot we as network engineers can learn from the study of networking in other domains. This is particularly true when thinking about how protocols are designed and deployed, and why some are notoriously hard to change.

So what are Robustness, Fragility, and Complexity, and how can they inform design and operation of our networks?

1 Why should this be?

BTW, what is a Cell/…/Ecosystem?

- A cell (tissue, organ, organism, ecology) is a network of compute and storage devices which is orchestrated by a set of elaborate multi-layer/multi-scale control systems.
  - Sound familiar?

- Design Principles of Biological Systems
  - Systems are built up of "simpler" components
  - Robustness
  - Optimality
  - Separation of time scales

- BTW, too abstract for Internet engineering?
  - Might think again…
  - See "IT'S ALIVE! IT'S ALIIIVE! Google's secretive Omega cloud acts like living thing – 'Biological' signals ripple through massive cluster management monster"
    - http://www.theregister.co.uk/2013/11/04/google_living_omega_cloud/

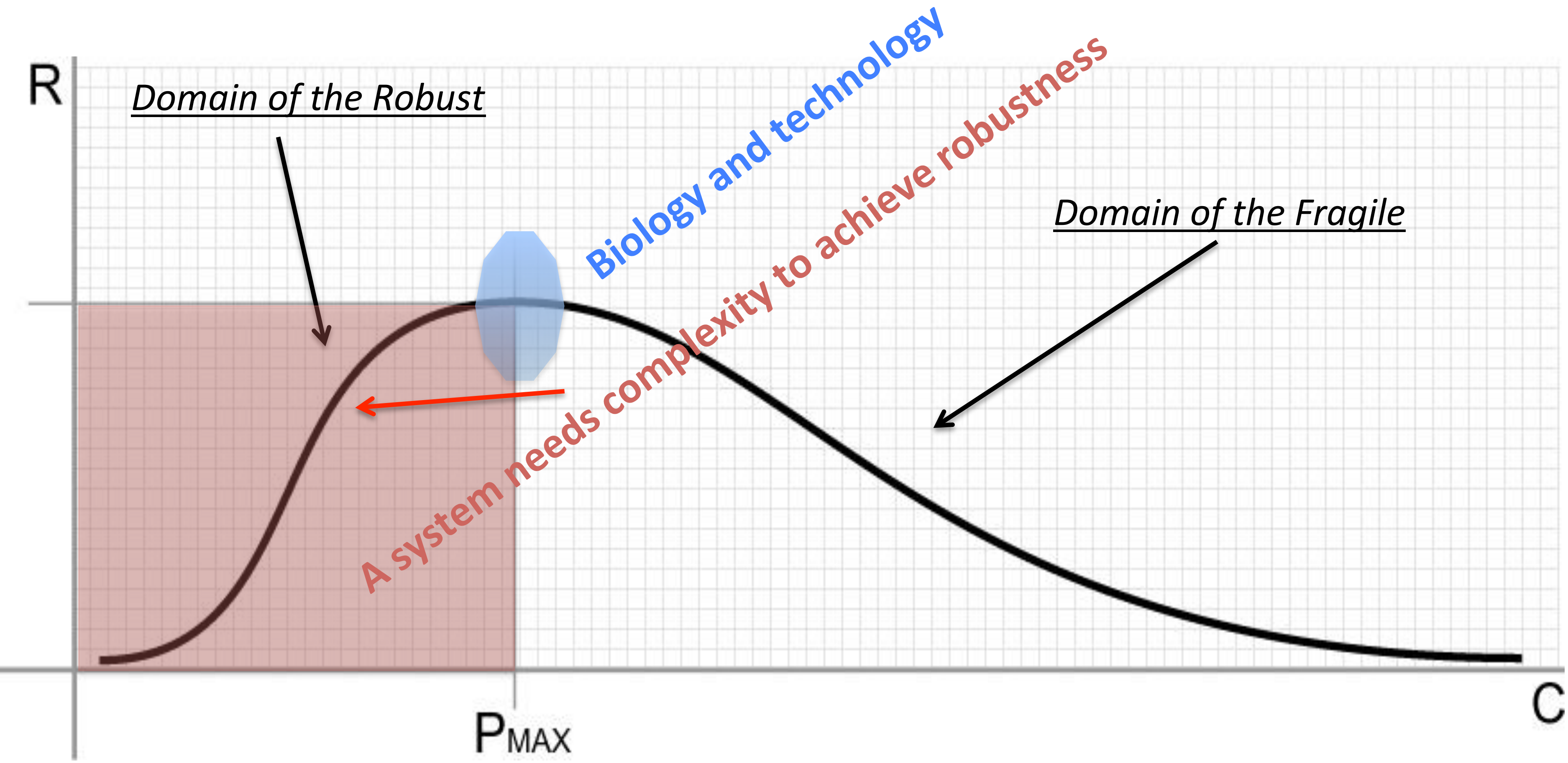
- How is this related to complexity?

Complexity is about Robustness to Uncertainty to Environment and Components

"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.***"

See Alderson, D. and J. Doyle, "Contrasting Views of Complexity and Their Implications For Network-Centric Infrastructures", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 40, NO. 4, JULY 2010

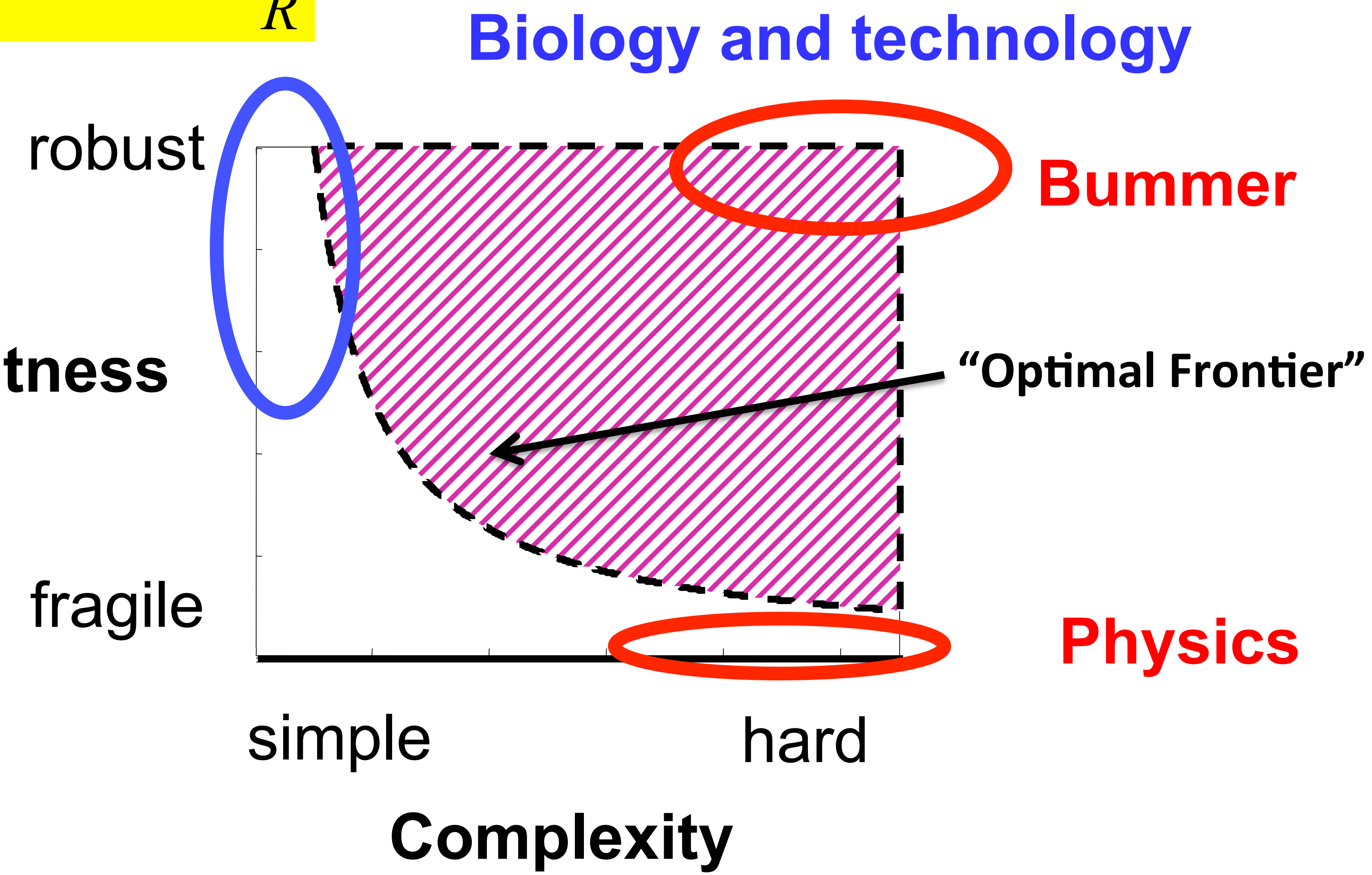
Robustness vs. Complexity



Increasing number of policies, protocols, configurations and interactions (well, and code)

Another way to look at the CR "Frontier"

(Toy) Theorem: $C \le \frac{1}{R}$



Original slide courtesy John Doyle

So what are Robustness and Fragility?

- Definition: A *[property*] of a *[system]* is **robust** if it is *[invariant]* with respect to a *[set of perturbations]*, *up to some limit*
  - Robustness is the preservation of a certain property in the presence of uncertainty in components or the environment
  - Systems Biology: Biological systems are designed such that their important functions are insensitive to the naturally occurring variations in their parameters.
    - Limits the number of designs that can actually work in the real environment

- **Fragility** is the opposite of robustness
  - If you're fragile you depend on 2nd order effects (acceleration) and the "harm" curve is concave
  - A little more on this later…

- 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***
  - Or the system may collapse if it experiences perturbations above a certain threshold (K-fragile)

- For 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
  - Another example: VRRP provides robustness to failure of a router/interface, but introduces fragilities in the protocol/implementation
  - Complexity/Robustness Spirals

See Alderson, D. and J. Doyle, "Contrasting Views of Complexity and Their Implications For Network-Centric Infrastructures", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 40, NO. 4, JULY 2010

RYF Behavior is Everywhere

Robust	Yet Fragile
☺ Efficient, flexible metabolism	☹ Obesity and diabetes
☺ Complex development	☹ Rich microbe ecosystem
☺ Immune systems	☹ Inflammation, Auto-Im.
☺ Regeneration & renewal	☹ Cancer
🗐 Complex societies	☠ Epidemics, war, …
🗐 Advanced Technologies	💣 Catastrophic failures

- "Evolved" mechanisms for robustness *allow for,* even *facilitate,* novel, severe fragilities elsewhere
- Often involving hijacking/exploiting the same mechanism
  - We've certainly seen this in the Internet space (consider DDOS of various varieties)
- These are hard constraints (i.e., RYF behavior is conserved)

Robustness is a Generalized System Feature

- **Scalability** is robustness to changes to the size and complexity of a system as a whole
- **Evolvability** is robustness of lineages to changes on various (usually long) time scales
- Other system features cast as robustness
  - **Reliability** is robustness to component failures
  - **Efficiency** is robustness to resource scarcity
  - **Modularity** is robustness to component rearrangements
- BTW, what are the features we're seeking from SDN/NFV/Cloud?

Fragility and Scaling are Related

- A bit of a formal description of fragility
  - Let *z* be some stress level, *p* some property, and
  - Let *H(p,z)* be the (negative valued) harm function
  - Then for the fragile the following must hold
    - $H(p,nz) < nH(p,z)$ **for** $0 < nz < K$
- For example, a coffee cup on a table suffers non-linearly more from large deviations (*H(p, nz)*) than from the cumulative effect of smaller events (*nH(p,z)*)
  - So the cup is damaged far more by *tail events* than those within a few σ of the mean
  - Too theoretical? Perhaps, but consider: ARP storms, micro-loops, congestion collapse, AS 7007, …
  - BTW, nature requires this property
    - Consider: jumping off something 1 foot high 30 times vs. jumping off something 30 feet high once
- When we say something scales like $O(n^2)$, what we mean is the damage to the network has constant acceleration (2) for *weird* enough n (e.g., outside say, 5 σ)
  - Again, ARP storms, congestion collapse, AS 7007, DDOS, … → non-linear damage

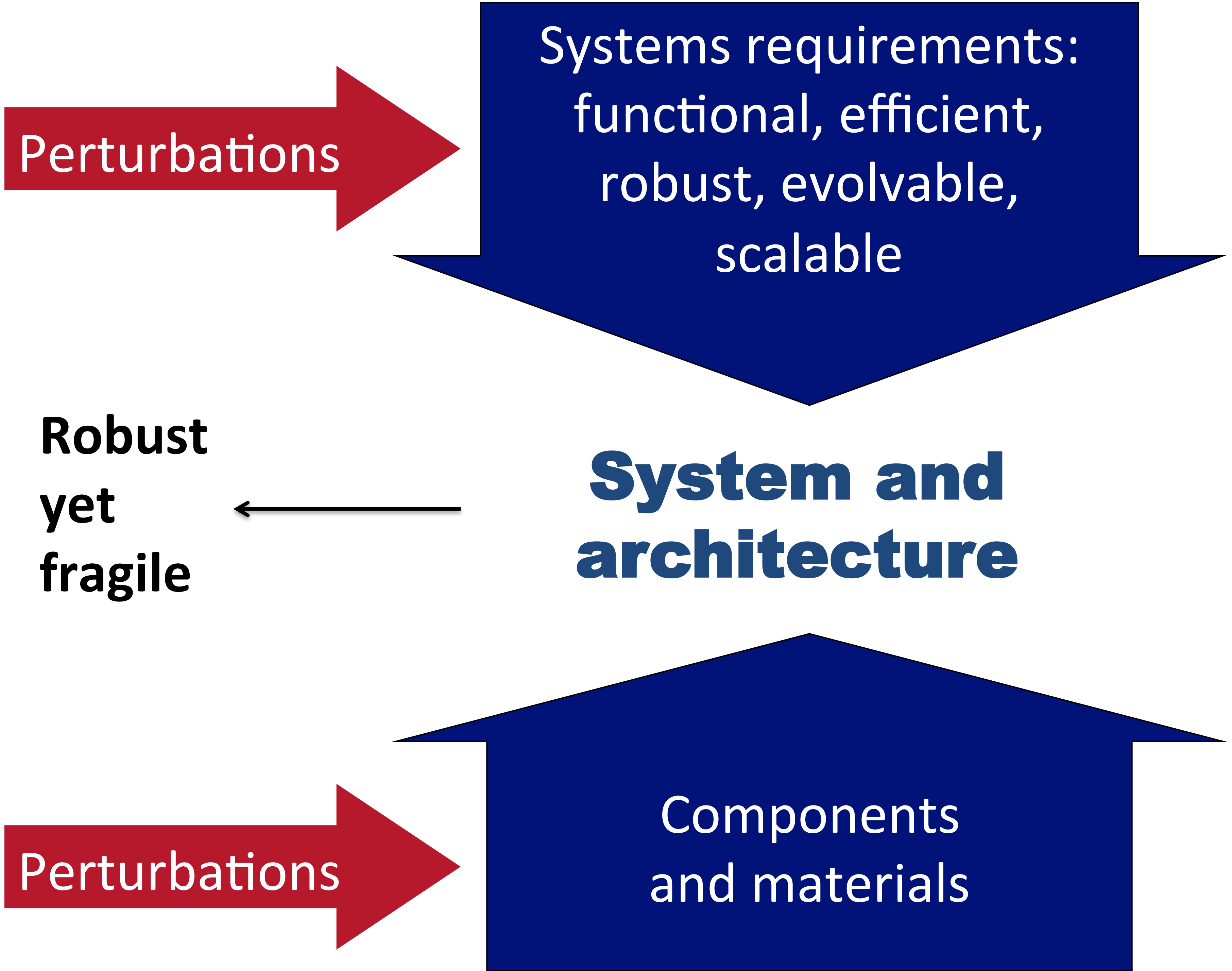
Idea courtesy Nassim Taleb, see http://www.fooledbyrandomness.com/

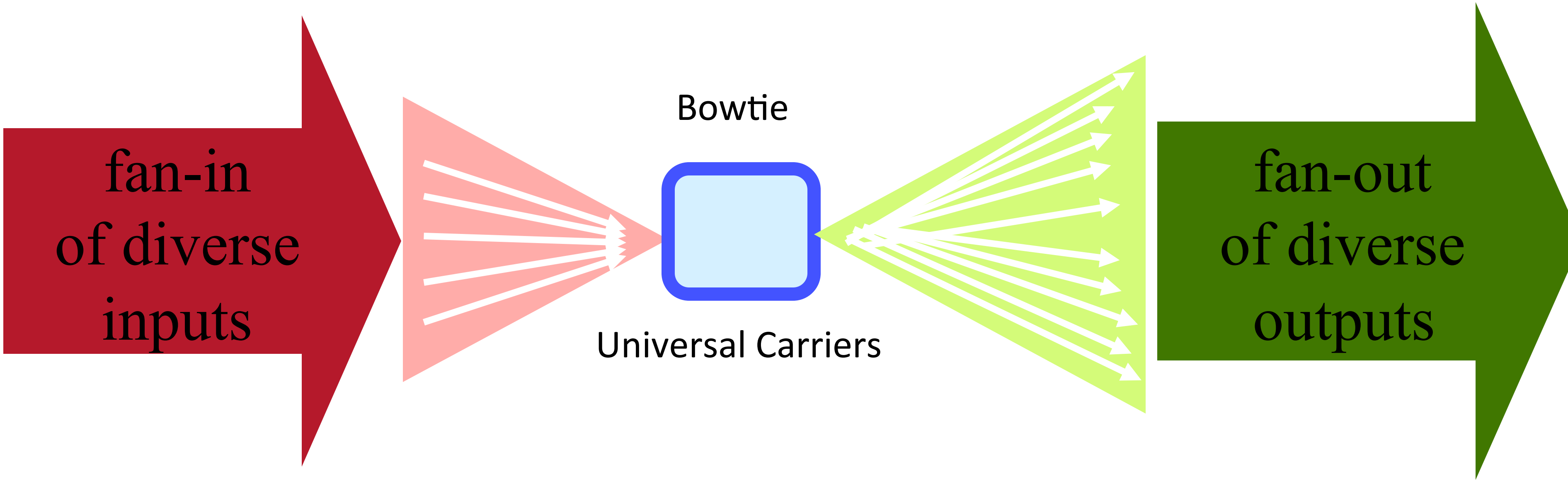
Understanding RYF is **The** Challenge

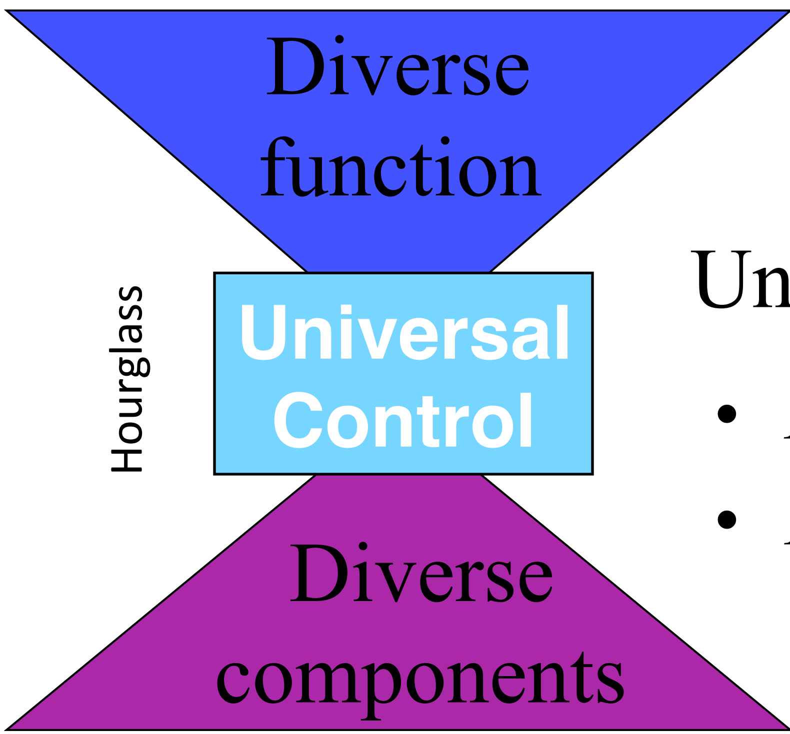
- Turns out that managing/understanding RYF behavior is ***the most essential challenge*** in technology, society, politics, ecosystems, medicine, etc. This means…
  - Managing spiraling complexity/fragility
  - Not predicting what is likely or typical
  - But rather understanding what is catastrophic (though perhaps rare)
  - → ***understanding the hidden nature of complexity***

- BTW, it is much easier to create the robust features than it is to prevent the fragilities
  - In addition, there are poorly understood "conservation laws" at work

- Thesis redux: Our technology (Internet) has reached a stage of ubiquity and complexity such that we face these same challenges
  - And what is the effect of s/w oriented technologies s/a SDN/NFV/Cloud?
  - Increasing feedback, variability, spiraling complexity → *Classic* RYF
    - and the nature of software itself (and how we build it)


Systems requirements: functional, efficient, robust, evolvable, scalable
Perturbations
Robust yet fragile
System and architecture
Perturbations
Components and materials





Universal Architectural Principles

- *Hourglasses* for layering of control
- *Bowties* for flows within layers

Slide courtesy John Doyle

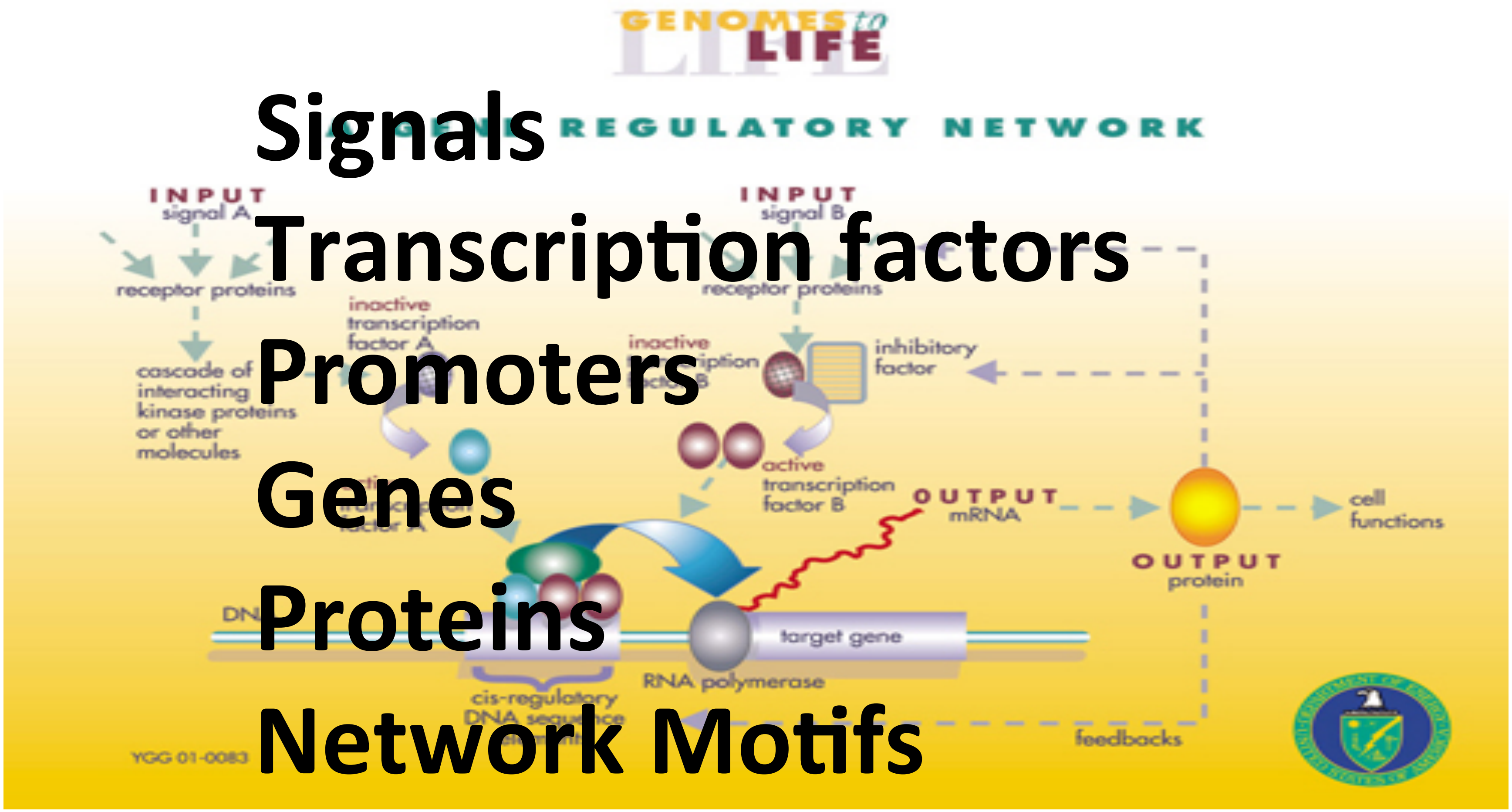
Biology and Technology Show Remarkable Convergence

- Biology and Advanced Technology differ widely at the molecular and device levels
- However, they show remarkable convergence at higher layers of organization
  - Recall Weaver's classification of problems
  - *Convergent Evolution*
    - Fruit fly, bird, bat, 747 → Wings, lift and control might be a good idea if you want to fly
- What we have learned is that there are ***universal architectural building blocks*** found in systems that scale and are evolvable. More on this later.
- A couple of examples
  - Transcription Networks and Biological vs. Engineered Circuits
  - Bowties/Hourglasses

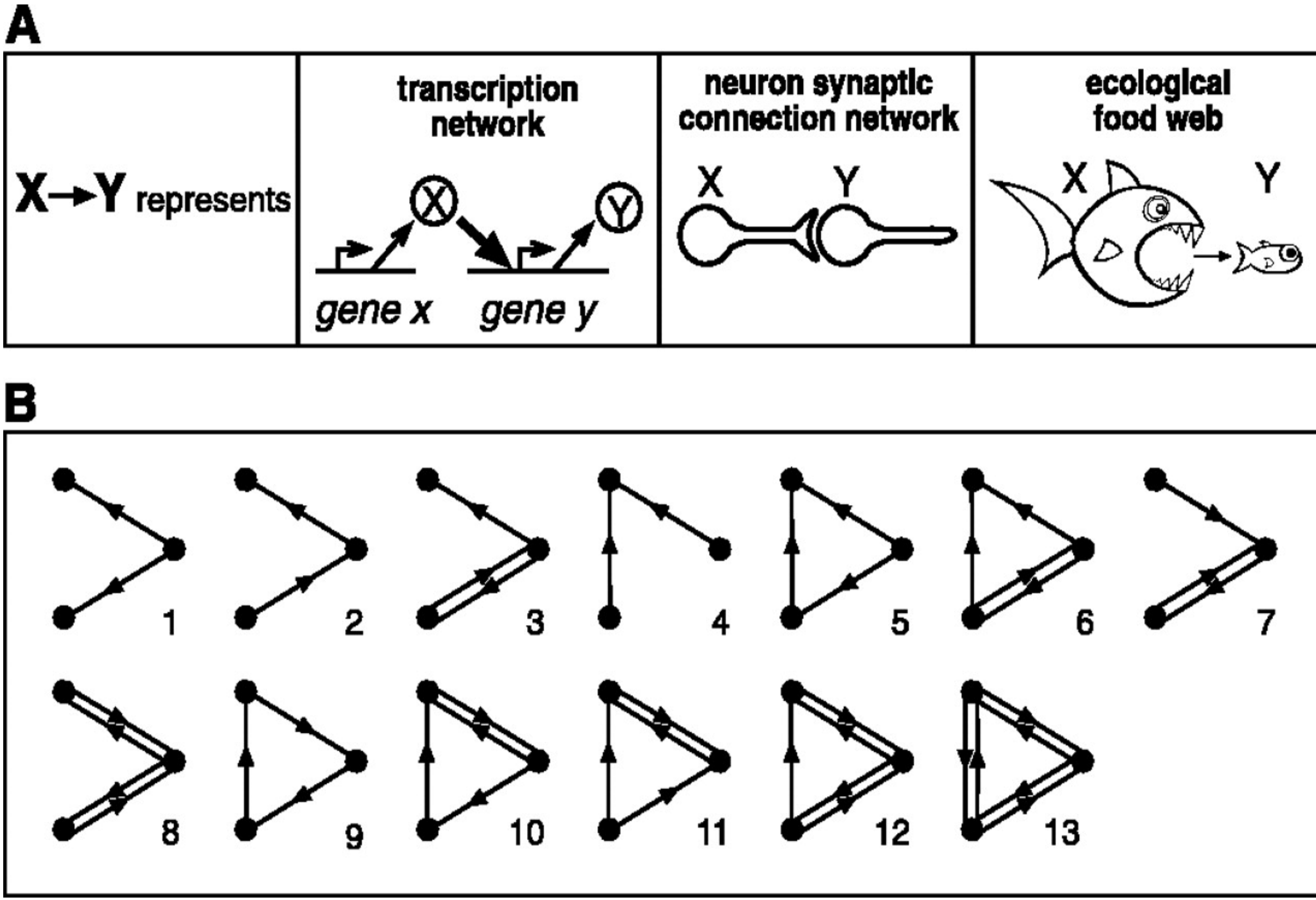
See Marie E. Csete and John C. Doyle, "Reverse Engineering of Biological Complexity", http://www.cds.caltech.edu/~doyle/wiki/images/0/05/ScienceOnlinePDF.pdf

Biological (Transcription) Networks

Signals
Transcription factors
Promoters
Genes
Proteins
Network Motifs



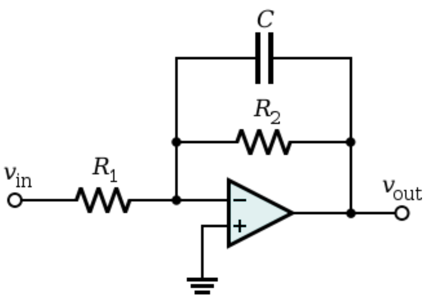
Network Motifs
There are 13 three node motifs



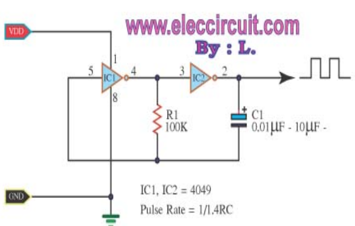
Feed Forward Loops

Technology

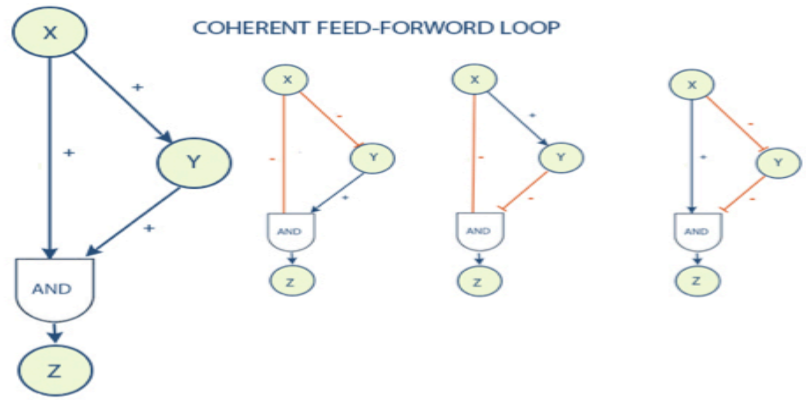
Low Pass Filter

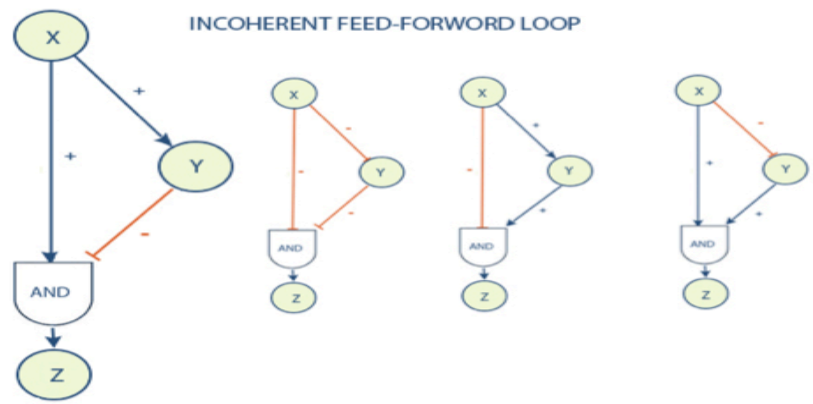


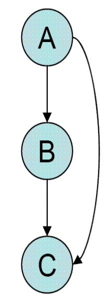
Pulse Generator



Biology

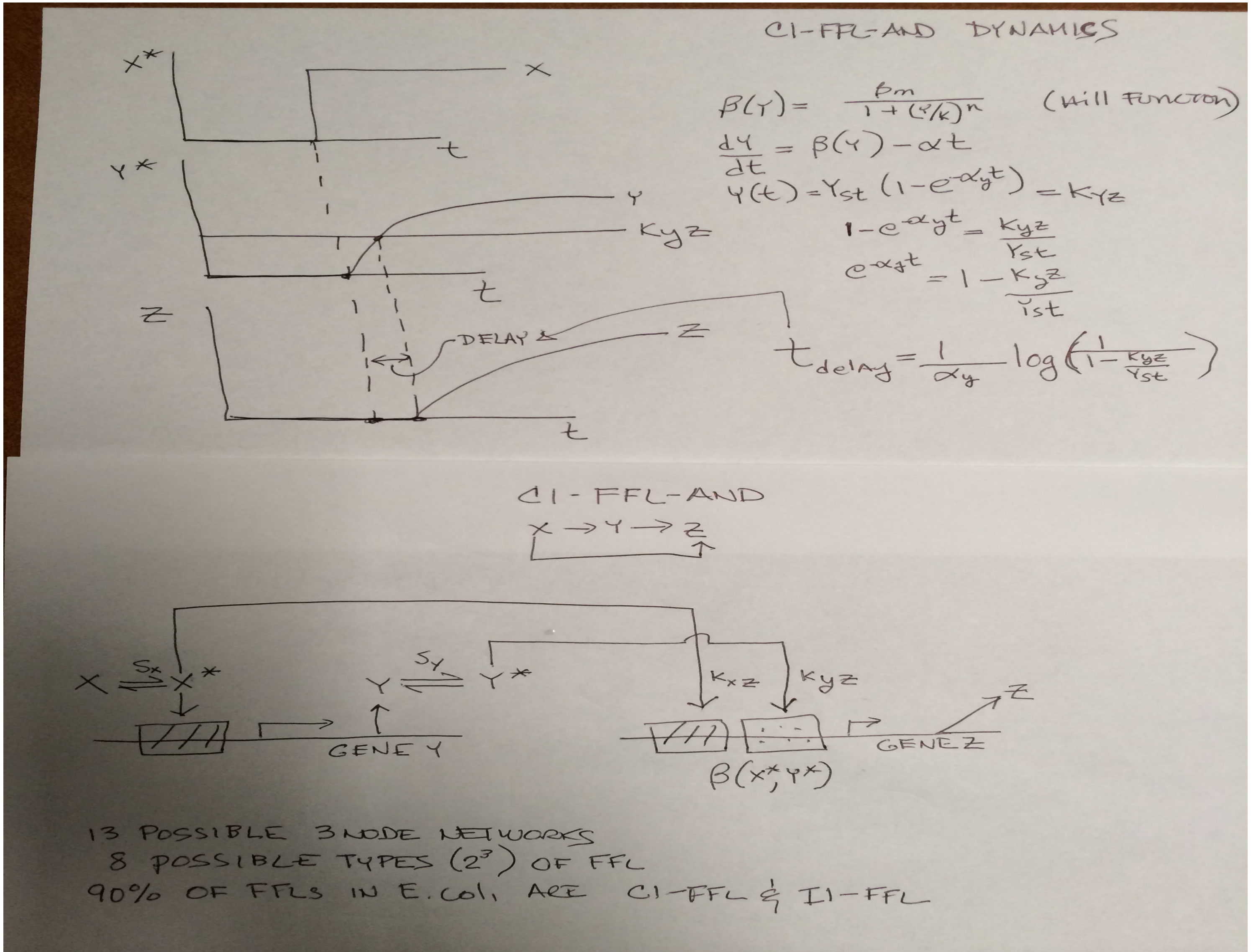






See Alon, U., et.al. "Coherent FeedForward Loops Serves as a Sign-Sensitive Delay Element in Transcription Networks", http://ender.bu.edu/~tgardner/be567_readings/Cell signalling/Feedforward/Mangan.S_JMB_03.pdf

How Does This Actually Work?
(on step)

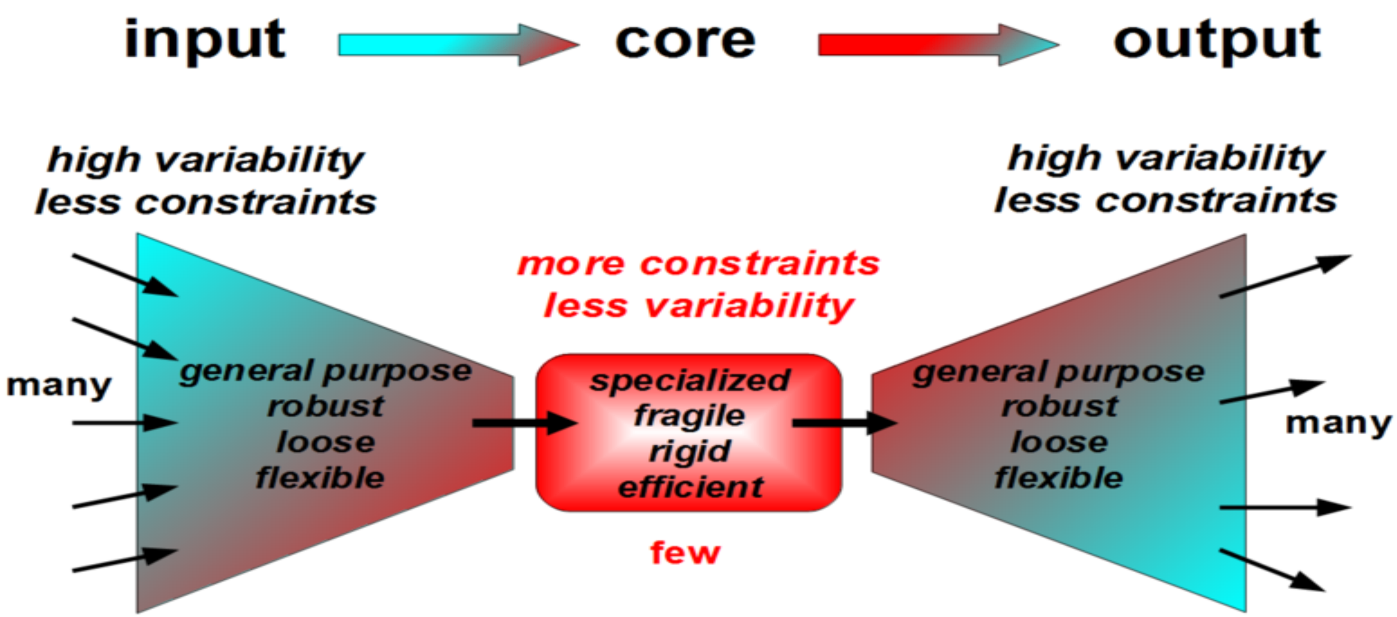


Universal Architectural Building Blocks

- What we have learned is that there are ***universal architectural building blocks*** found in systems that scale and are evolvable. These include
  - **RYF complexity**
  - **Bowtie/Hourglass architectures**
  - Protocol Based Architectures
  - Massively distributed with *robust* control loops
    - Contrast optimal control loops and hop-by-hop control
  - Highly layered
    - But with layer violations, e.g., Internet, overlay virtualization
  - Degeneracy

Bowties 101

Constraints that Deconstrain
Schematic of a "Layer"



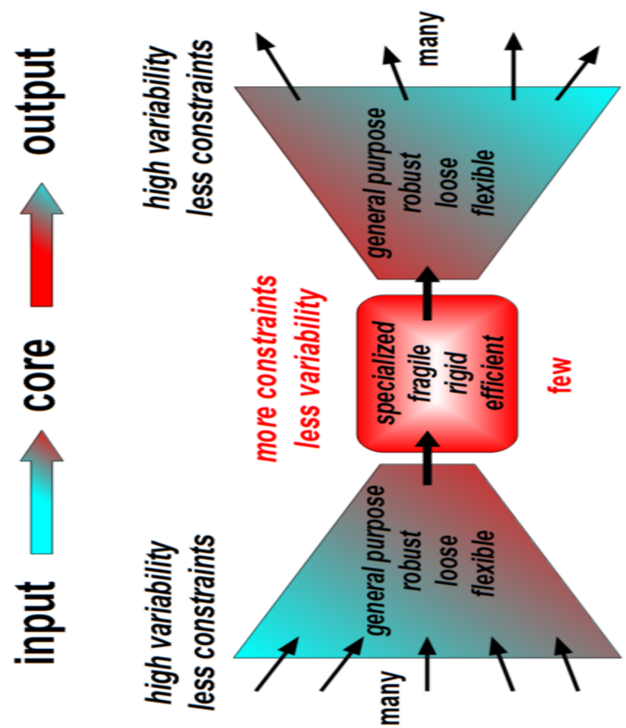
For example, the reactions and metabolites of core metabolism, e.g., *ATP metabolism*, Krebs/Citric Acid Cycle, … form a "metabolic knot". That is, ATP is a ***Universal Carrier*** for cellular energy.

1. Processes L-1 information and/or raw material flows into a "standardized" format (the L+1 abstraction)
2. Provides plug-and-play modularity for the layer above
3. Provides robustness but at the same time fragile to attacks against/using the standardized interface

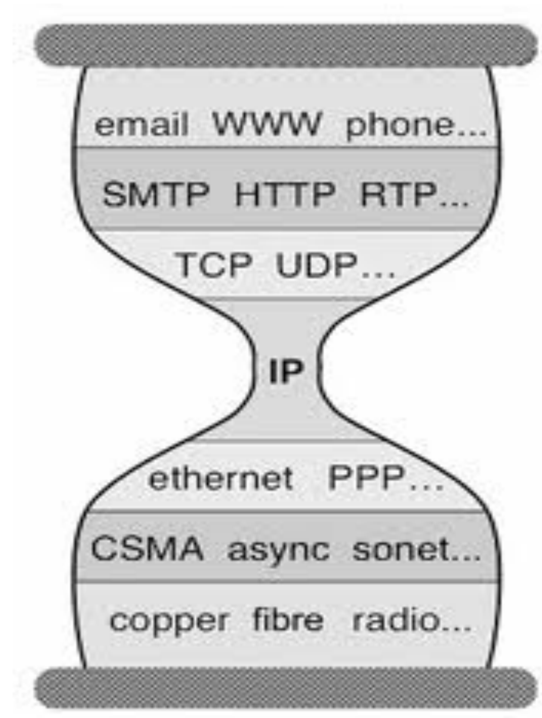
See Kirschner M., and Gerhart J., "Evolvability", Proc Natl Acad Sci USA , 95:8420–8427, 1998.

But Wait a Second

Anything Look Familiar?



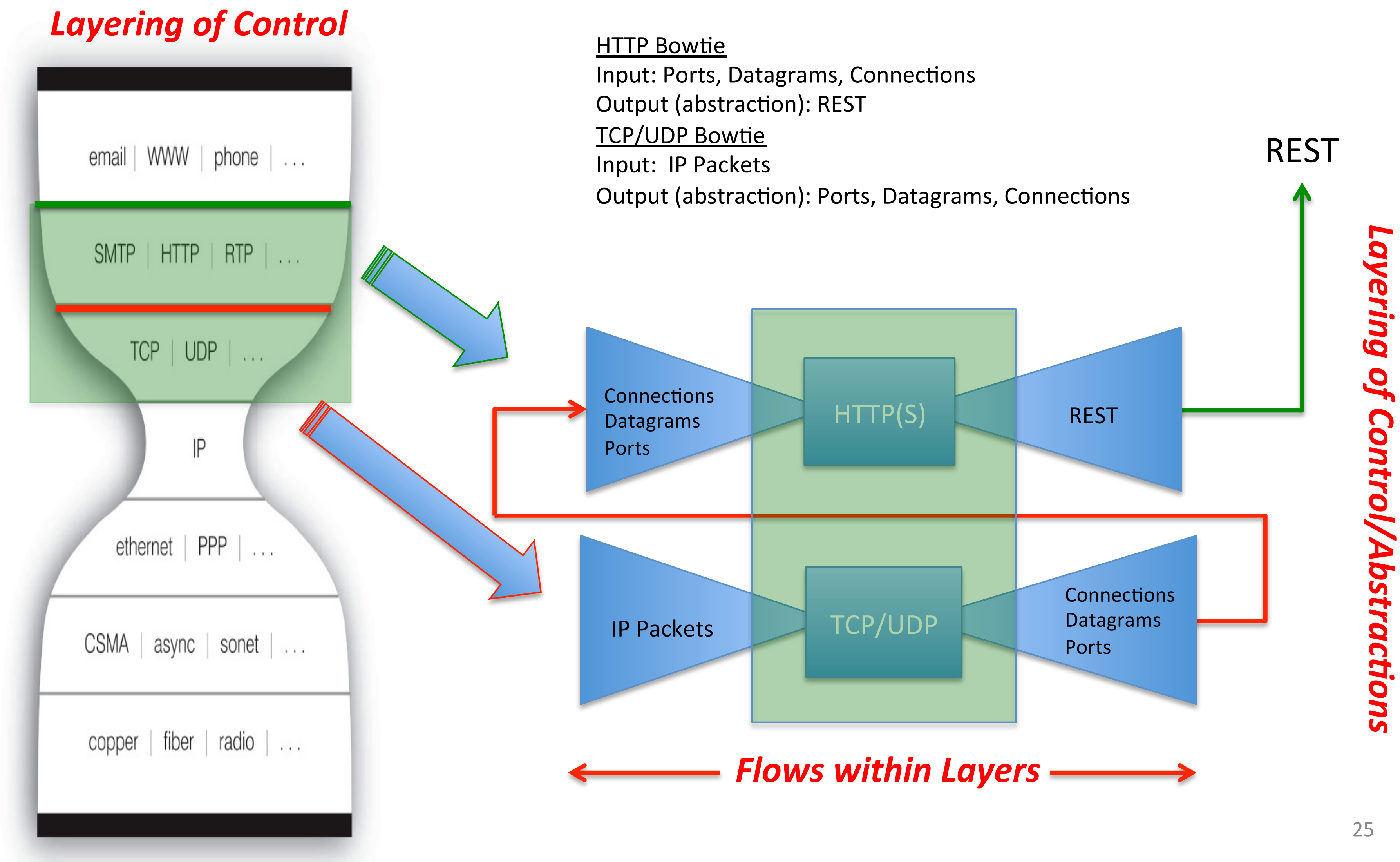
Bowtie Architecture



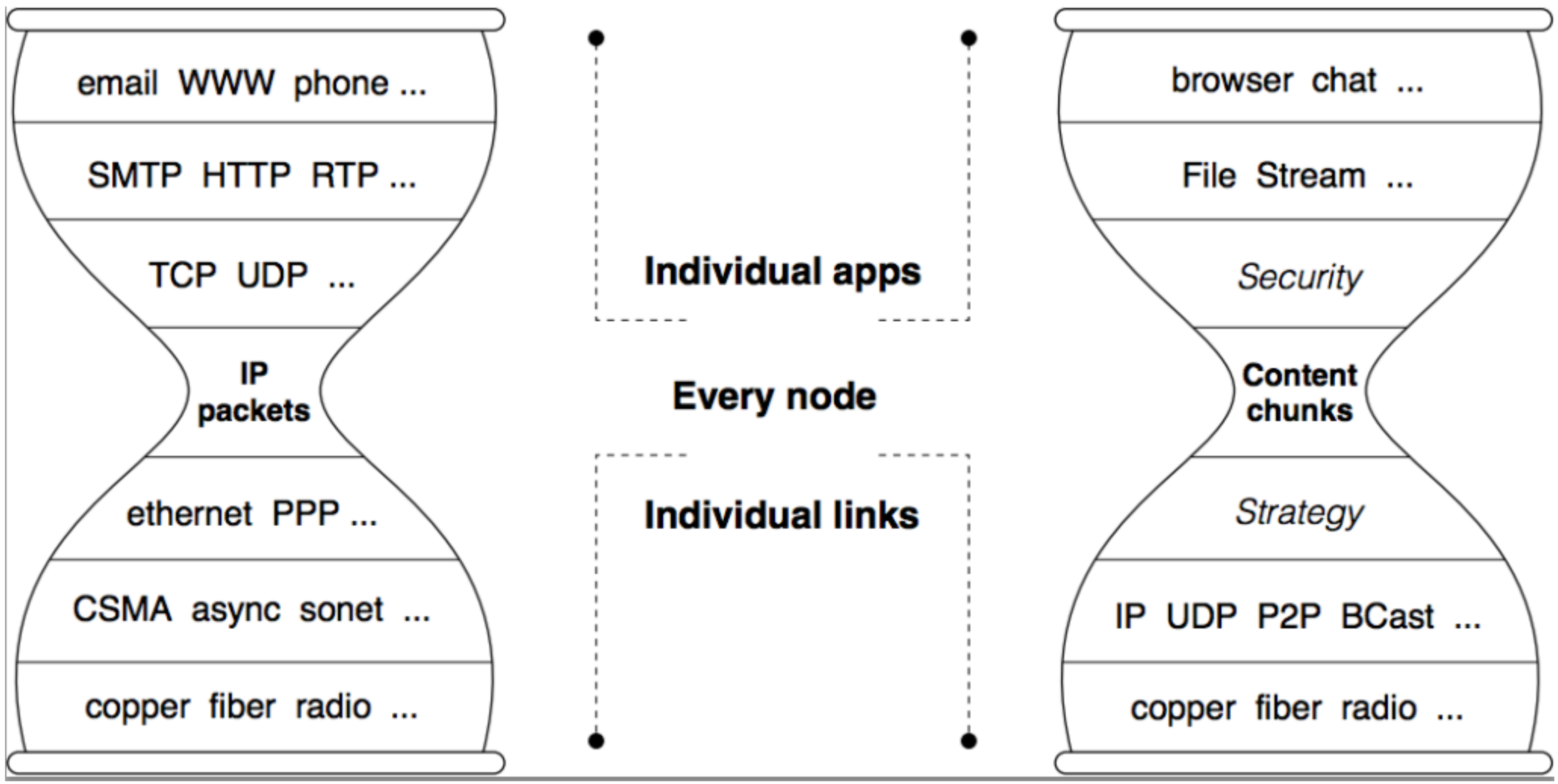
Hourglass Architecture

The *Protocol Hourglass* idea appears to have originated with Steve Deering. See Deering, S., “Watching the Waist of the Protocol Hourglass”, IETF 51, 2001, http://www.iab.org/wp-content/IAB-uploads/2011/03/hourglass-london-ietf.pdf. See also Akhshabi, S. and C. Dovrolis, “The Evolution of Layered Protocol Stacks Leads to an Hourglass-Shaped Architecture”, http://conferences.sigcomm.org/sigcomm/2011/papers/sigcomm/p206.pdf.

The Nested Bowtie/Hourglass Architecture of the Internet



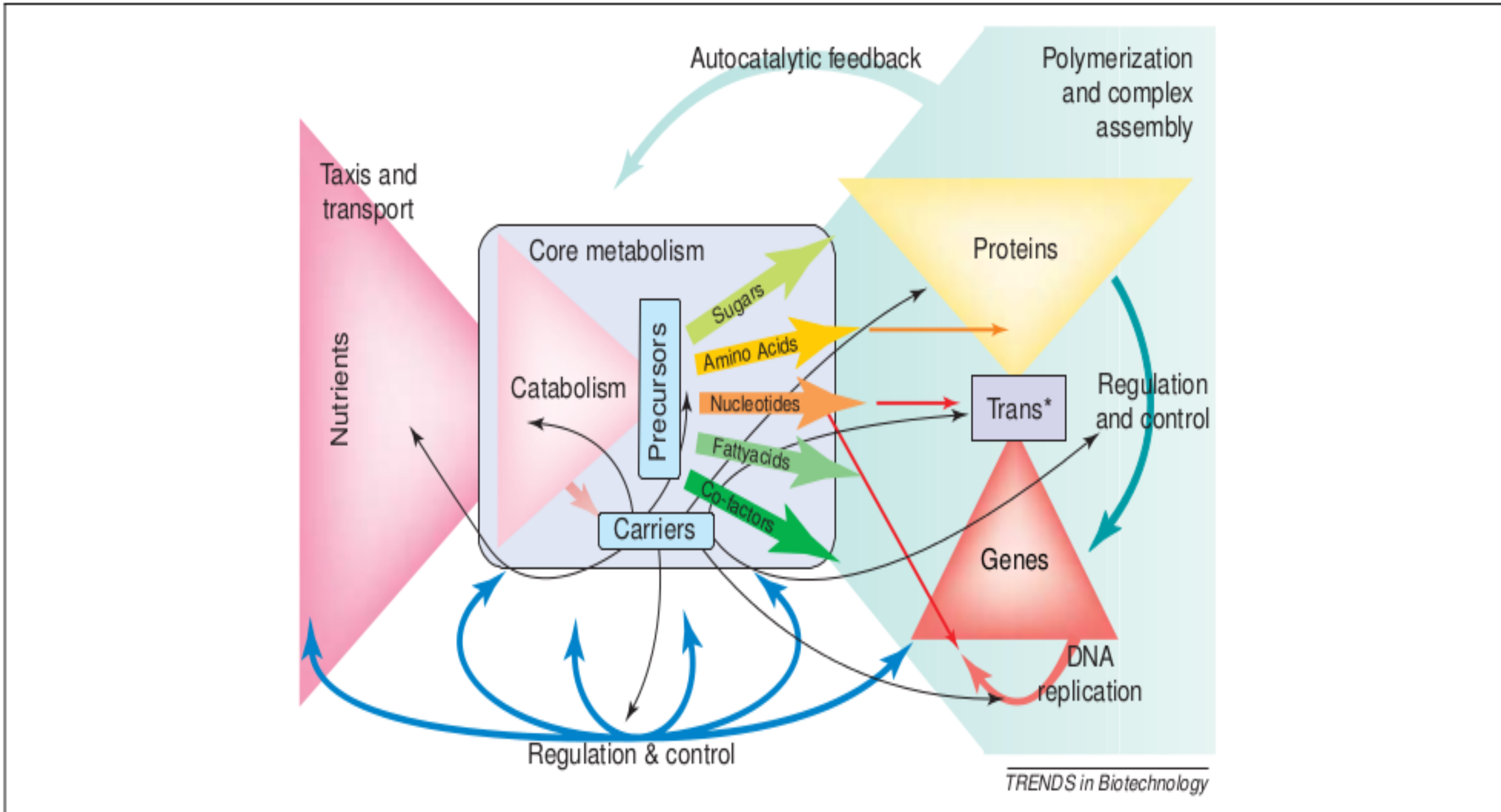
NDN Hourglass



See Named Data Networking, http://named-data.net/

In Practice Things are More Complicated

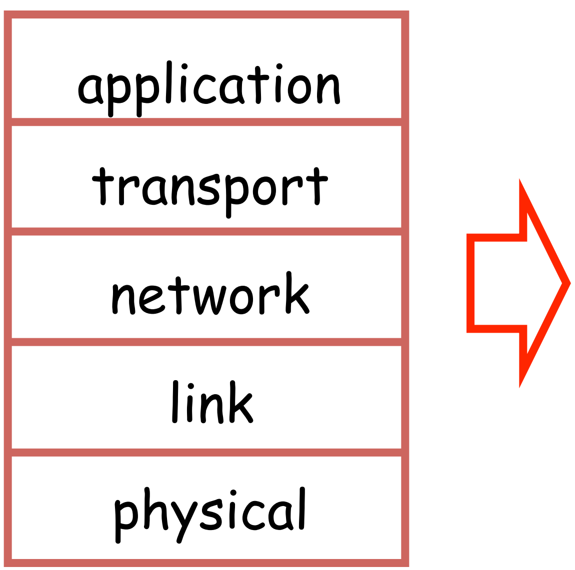
The Nested Bowtie/Hourglass Architecture of Metabolism



See Csete, M. and J. Doyle, “Bow ties, metabolism and disease”, TRENDS in Biotechnology, Vol. 22 No. 9, Sept 2004

BTW, we can look at layering as optimization decomposition

- Each layer is abstracted as an optimization problem
- Operation of a layer is a distributed solution
- Results of one problem (layer) are parameters of others
- Timescale separation (operate at different timescales)



$$\max_{x \geq 0} \quad \sum_i U_i(x_i)$$
$$\text{subj to} \quad Rx \leq c(p)$$
$$x \in X$$

Application: utility

Phy: power

IP: routing

Link: scheduling

So…Biology vs. the Internet

<u>Similarities</u>

- *Evolvable architecture*
- Robust yet fragile
- *Layering, modularity*
- Hourglass with bowties
- *Dynamics*
- *Feedback*
- Distributed/decentralized
- ***Not*** scale-free, edge-of-chaos, self-organized criticality, etc

<u>Differences</u>

- Metabolism
- Materials and energy
- **Autocatalytic feedback**
- Feedback complexity
- Development and regeneration
- > 3B years of evolution

An Autocatalytic Reaction is one in which at least one of the reactants is also a product. The rate equations for autocatalytic reactions are fundamentally non-linear.

Summary/Conclusions

- Robust systems "might be" intrinsically hard to understand
  - RYF complexity is an inherent property of complex technology
  - Software (e.g., SDN, NFV, Cloud, …) exacerbates the situation
  - And the Internet has reached an unprecedented level of complexity…
- Nonetheless, many of our goals for the Internet architecture revolve around how to achieve robustness…
  - which requires a deep understanding of the *necessary interplay between complexity and robustness, modularity, feedback, and fragility*
    - which is neither accidental nor superficial
  - Rather, architecture arises from "designs" to cope with uncertainty in environment and components
- Understanding these universal architectural features will help us achieve the scalability and evolvability (operability, understandability) we're seeking from the Internet architecture today
  - Multi-disciplinary approaches (e.g., Systems Biology) provide a template of how we might go about this.

Q&A

Thanks!