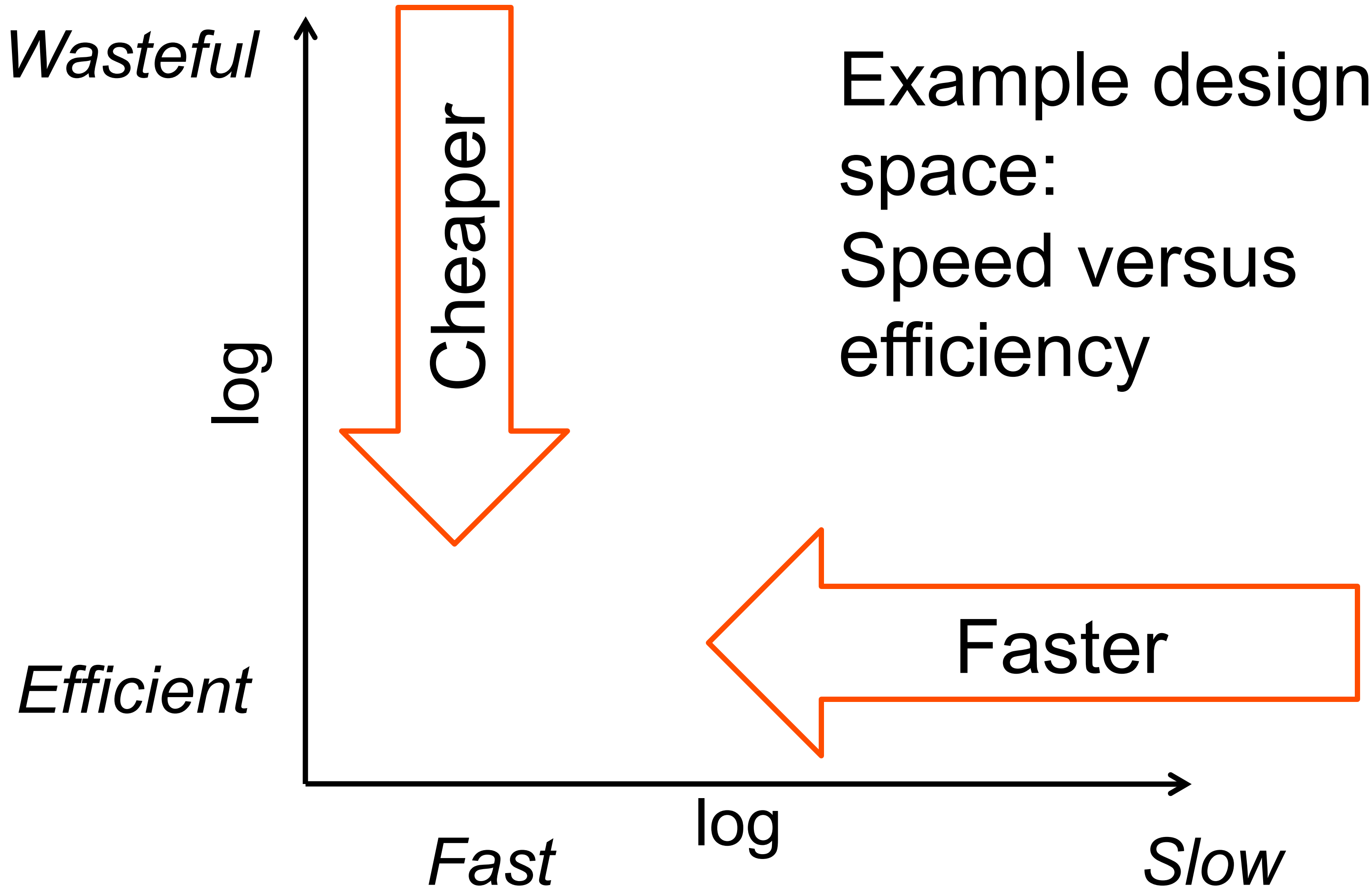
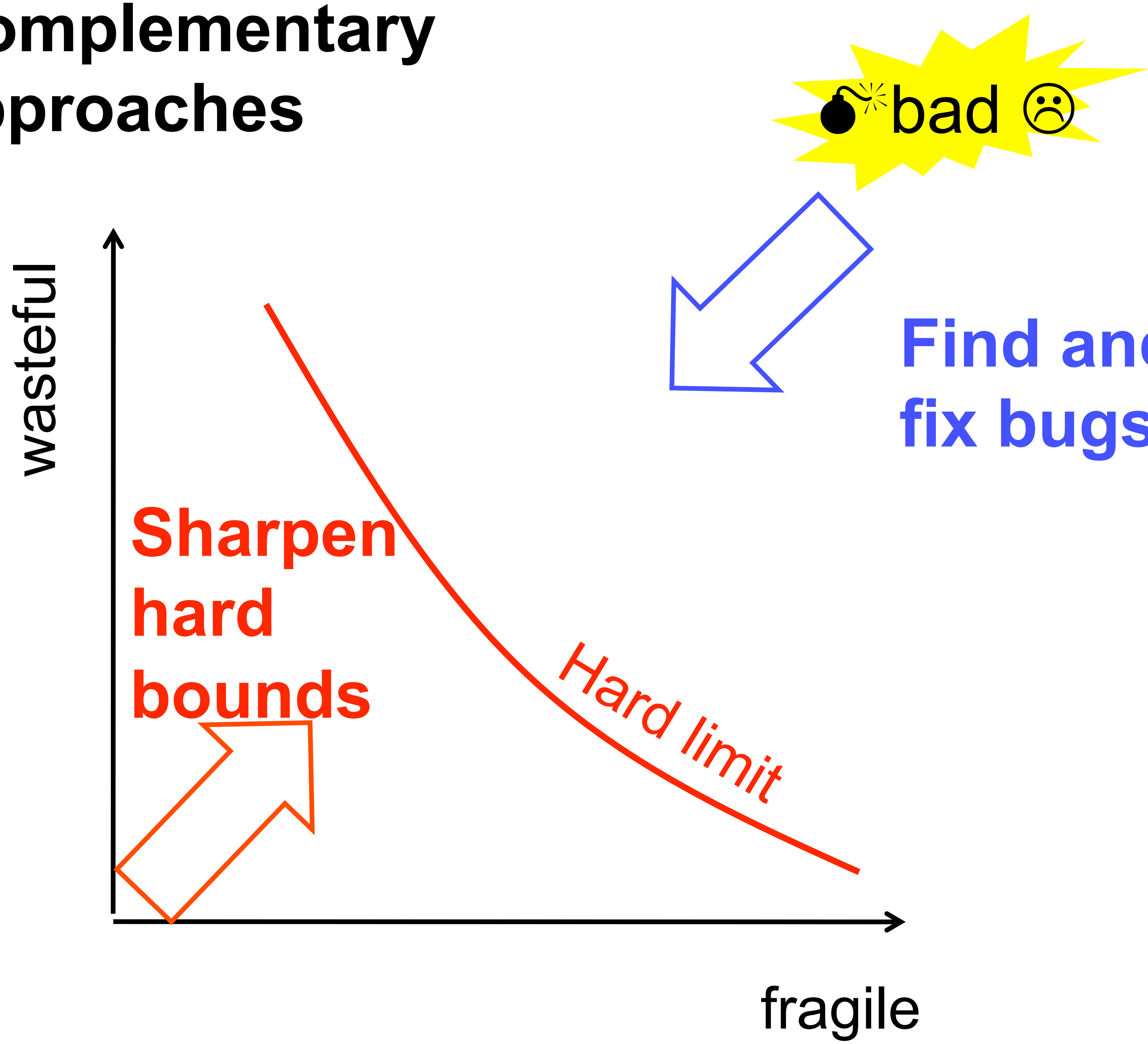
	M	**Tu**	**Wed**
10			
11			
12			
1			
2			
3			
4	?	CMI	

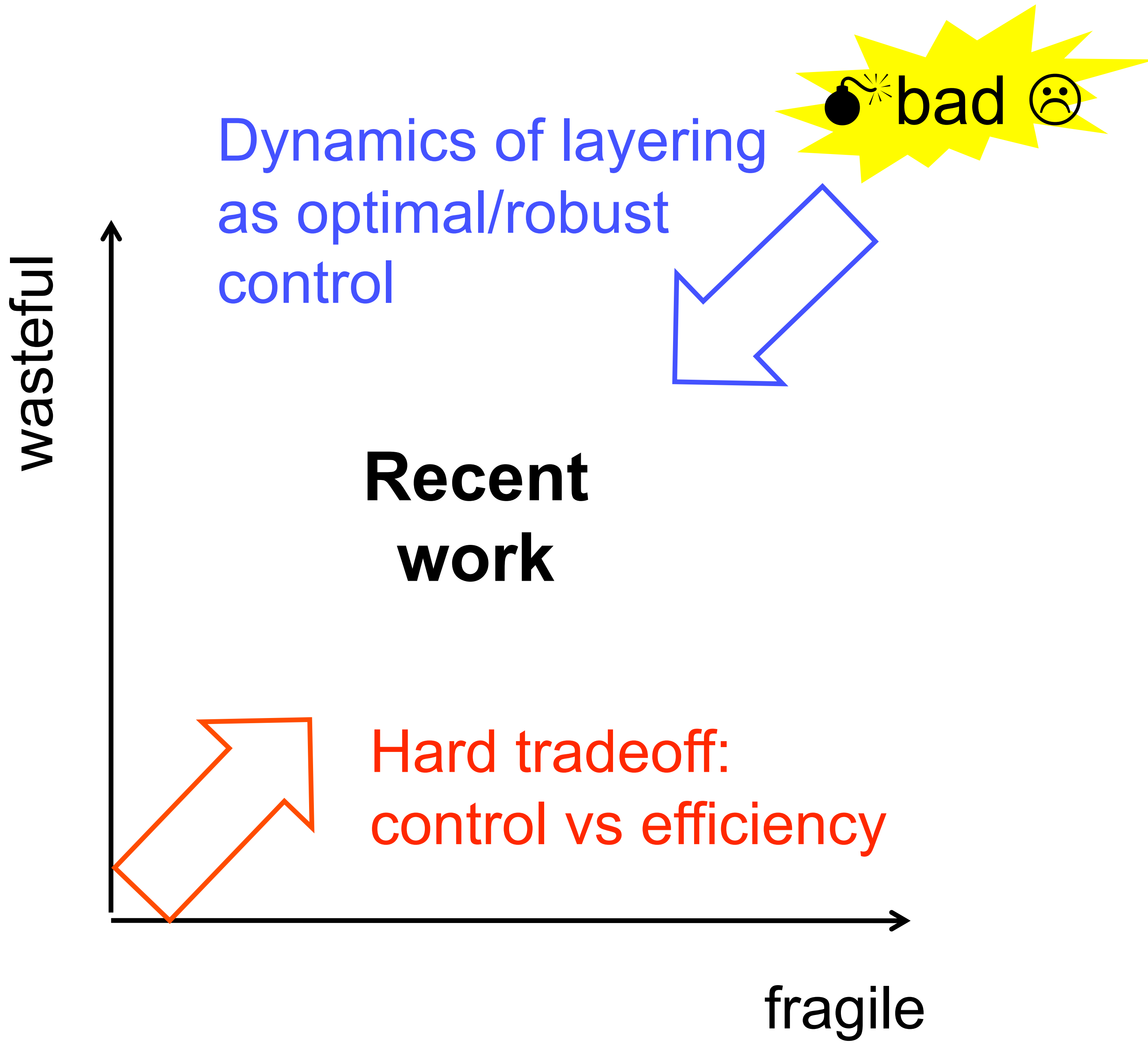
Design tradeoffs

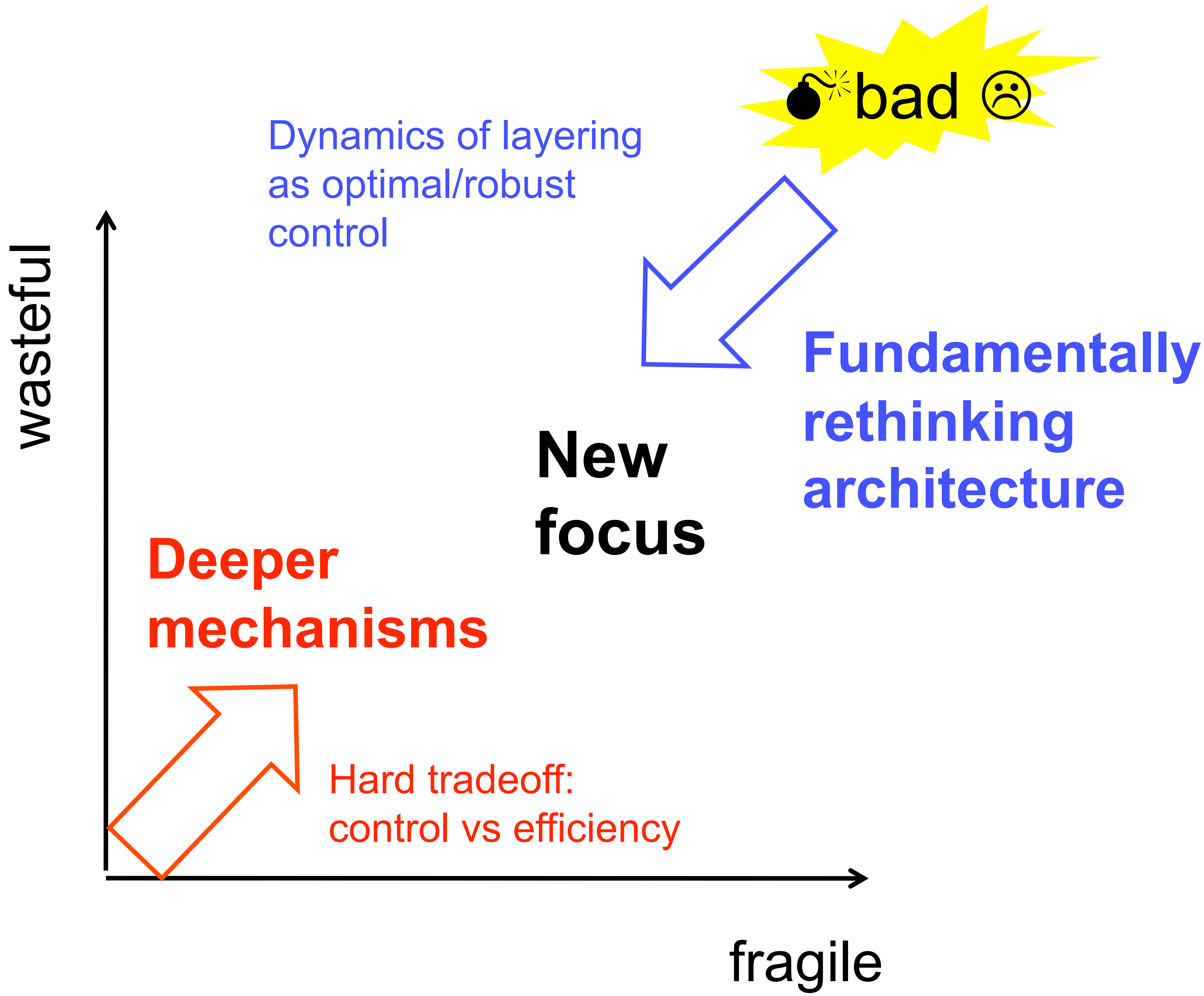


Example design space:
Speed versus efficiency

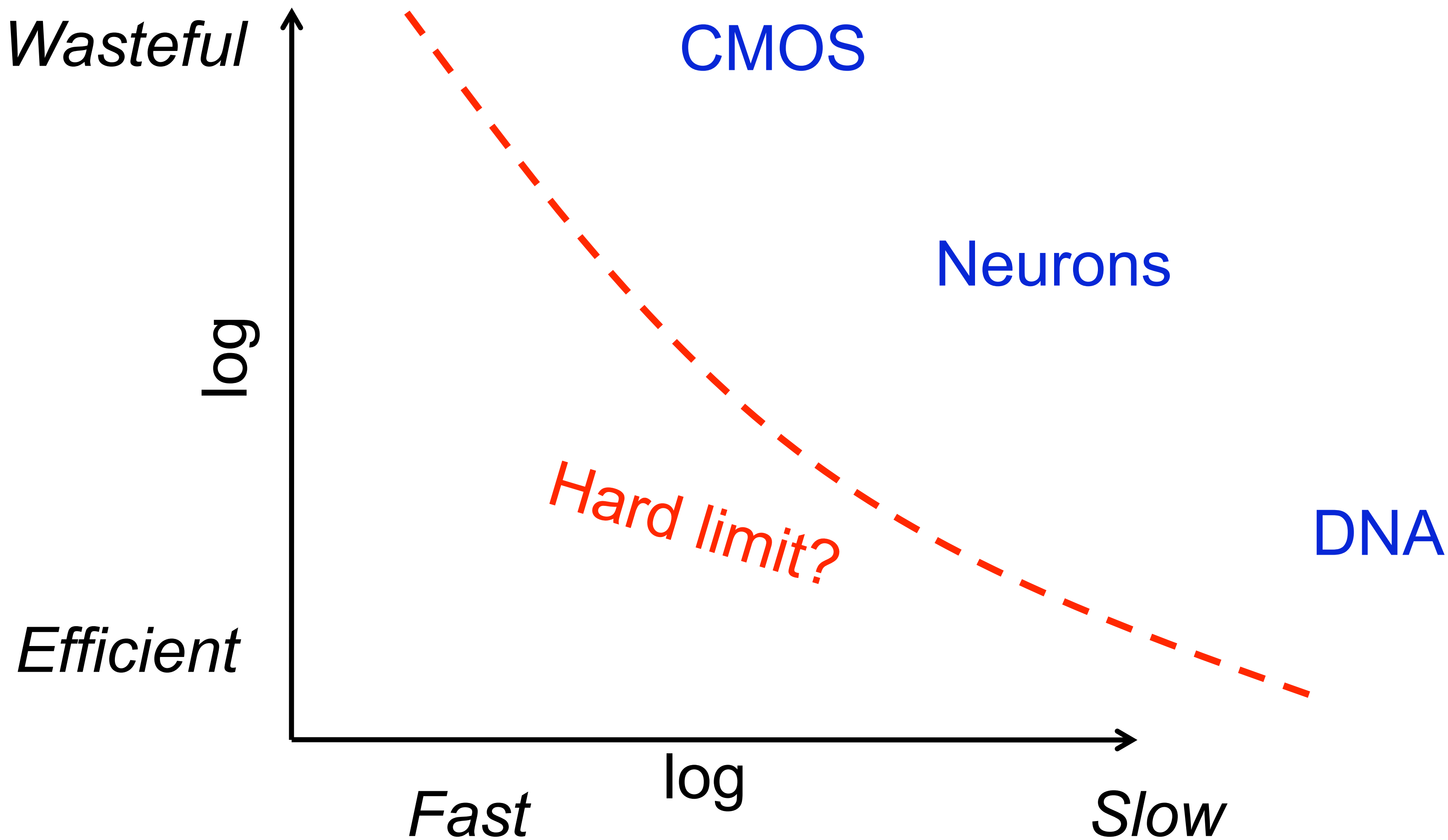
Complementary approaches




bad ☹
Dynamics of layering
as optimal/robust
control
wasteful
Recent
work
Hard tradeoff:
control vs efficiency
fragile

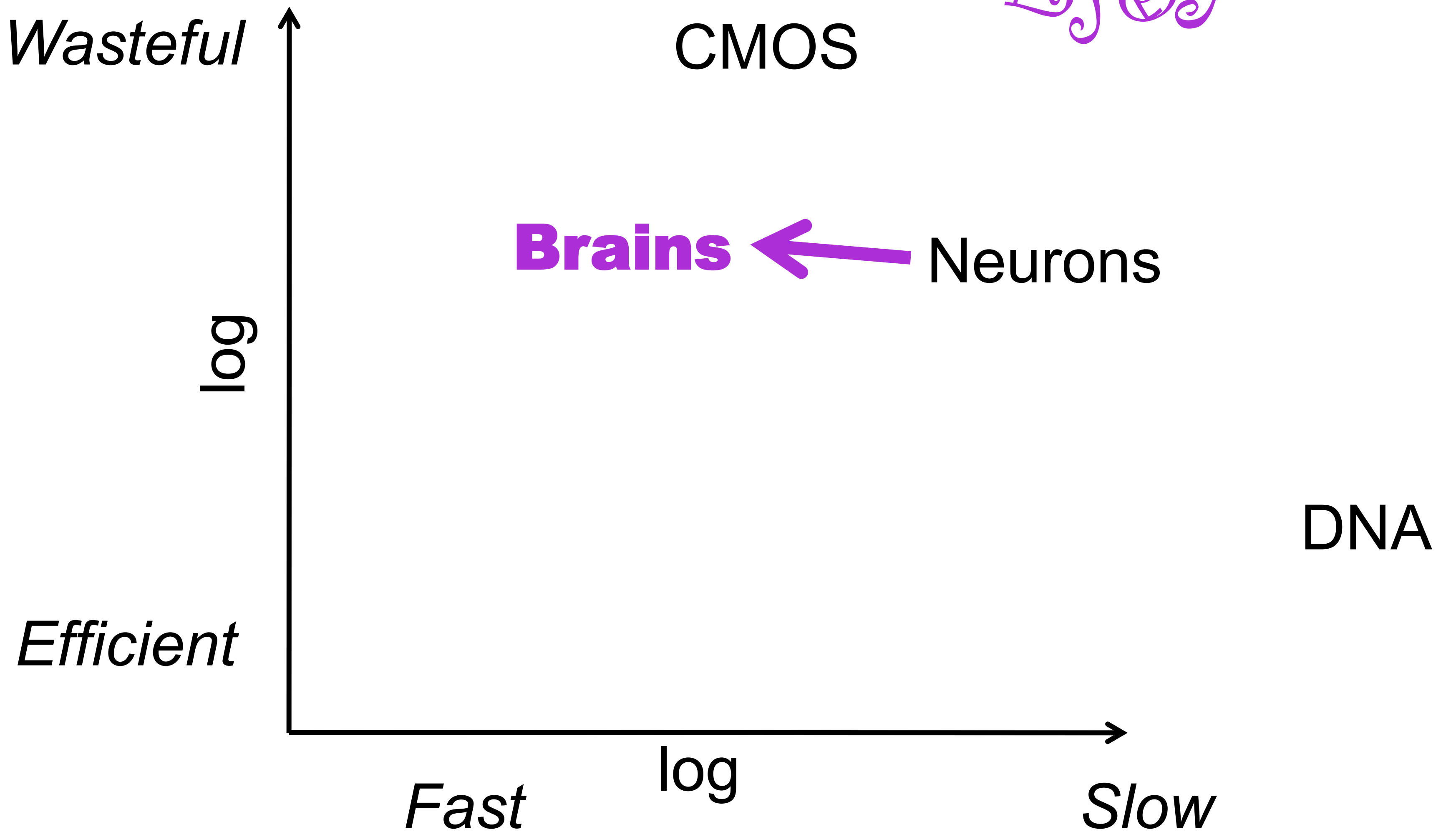

bad ☹
Dynamics of layering
as optimal/robust
control
wasteful
Fundamentally
rethinking
architecture
New
focus
Deeper
mechanisms
Hard tradeoff:
control vs efficiency
fragile

Computational hardware substrates

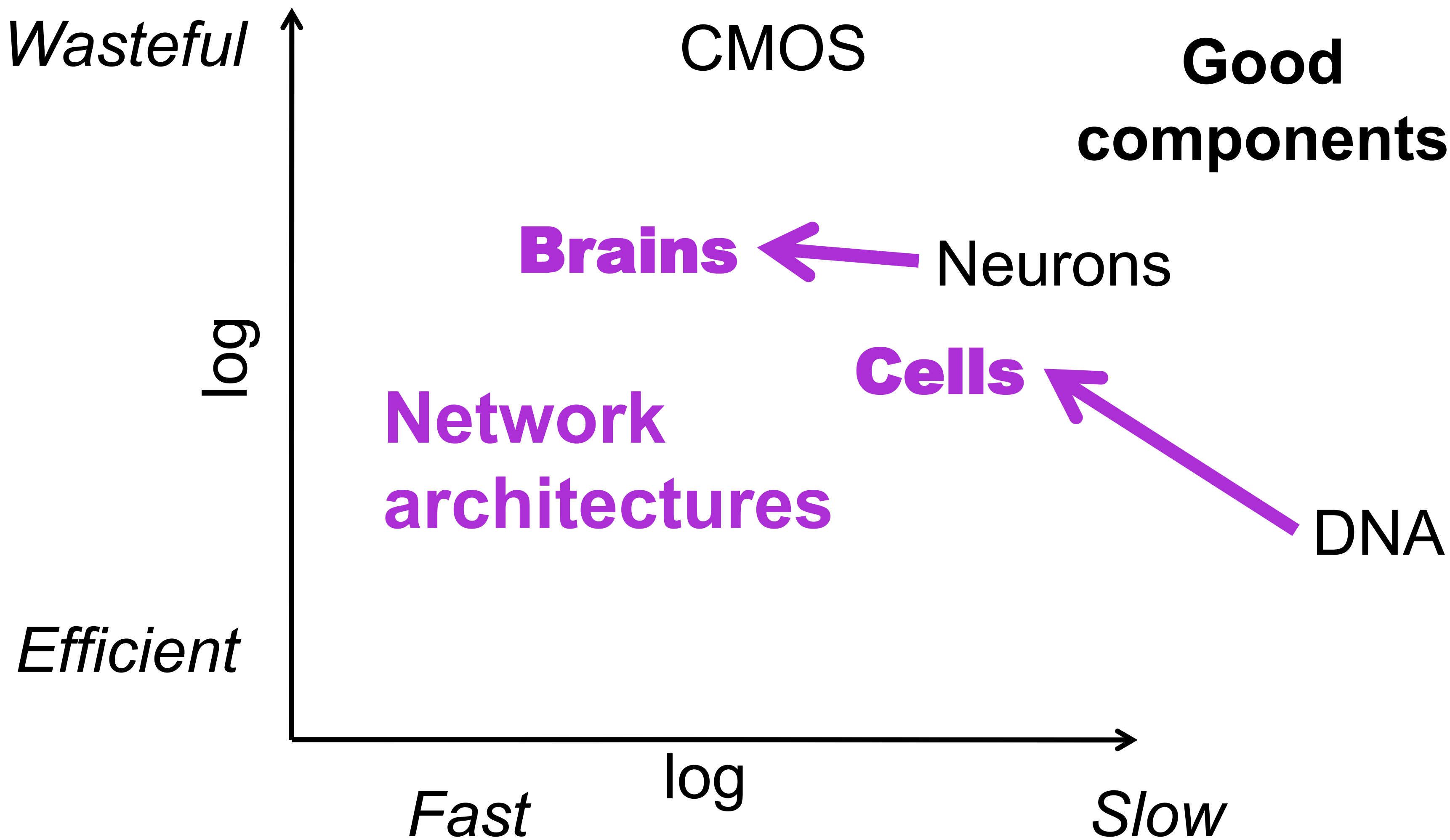


Some tasks: HARD for computers

EASY for us

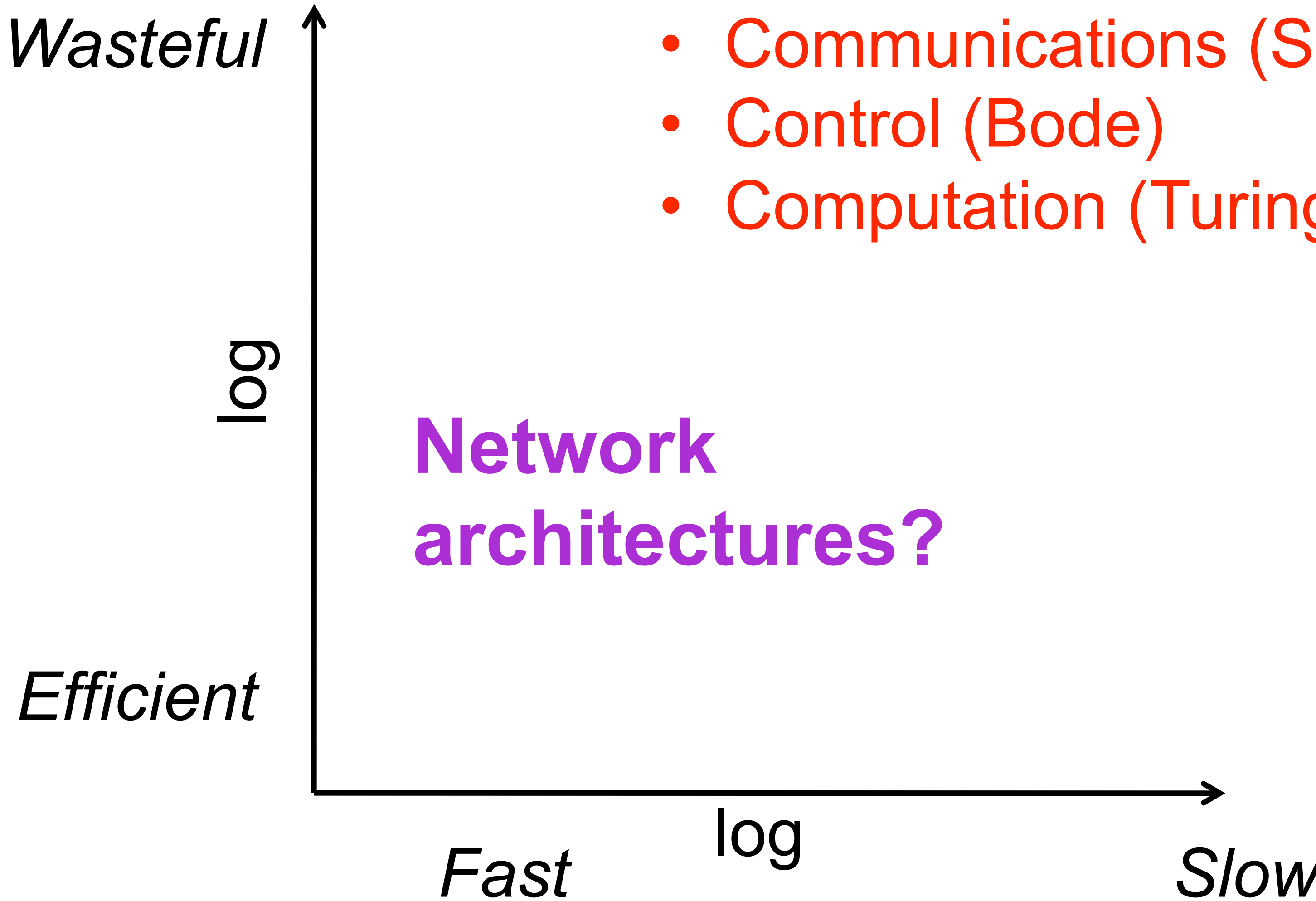


What makes this possible?



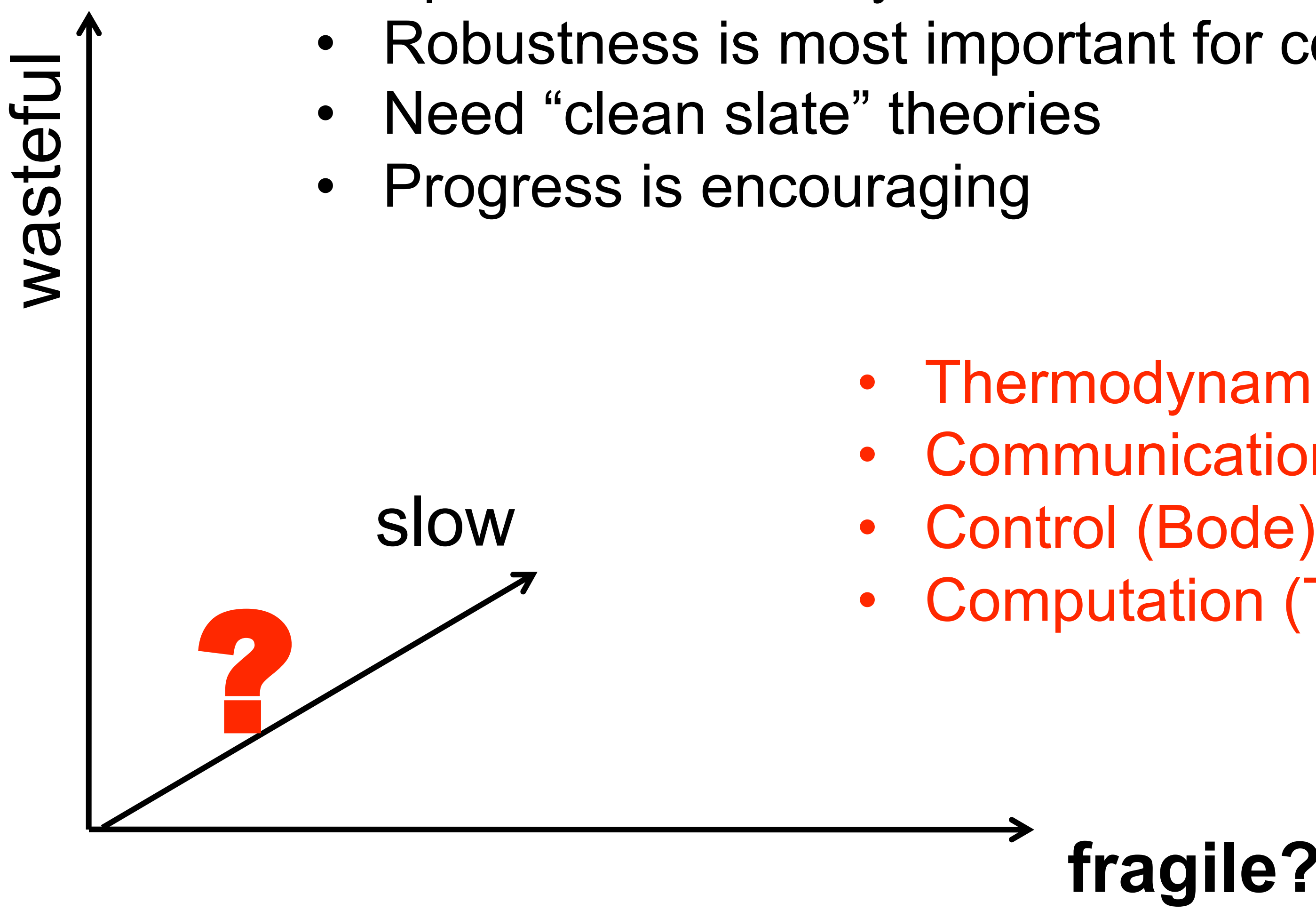
Standard theories are severely limited

- Thermodynamics (Carnot)
- Communications (Shannon)
- Control (Bode)
- Computation (Turing)



Standard theories are severely limited

- Each focuses on few dimensions
- Important tradeoffs are ***across*** these dimensions
- Speed vs efficiency vs robustness vs …
- Robustness is most important for complexity
- Need “clean slate” theories
- Progress is encouraging



- Thermodynamics (Carnot)
- Communications (Shannon)
- Control (Bode)
- Computation (Turing)

Most dimensions are robustness

Collapse for visualization

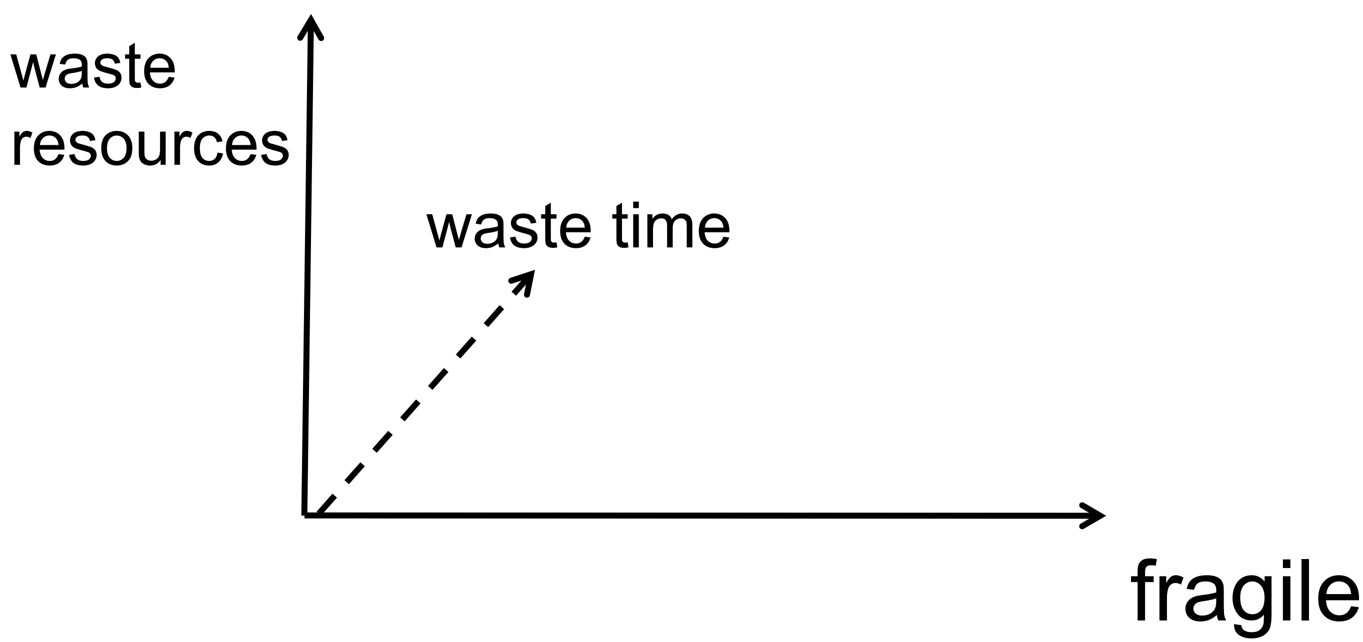
Robust
- Secure
- Scalable
- Evolvable
- Verifiable
- Maintainable
- Designable
- …

Fragile
- Not …
- Unverifiable
- Frozen
- …

fragile

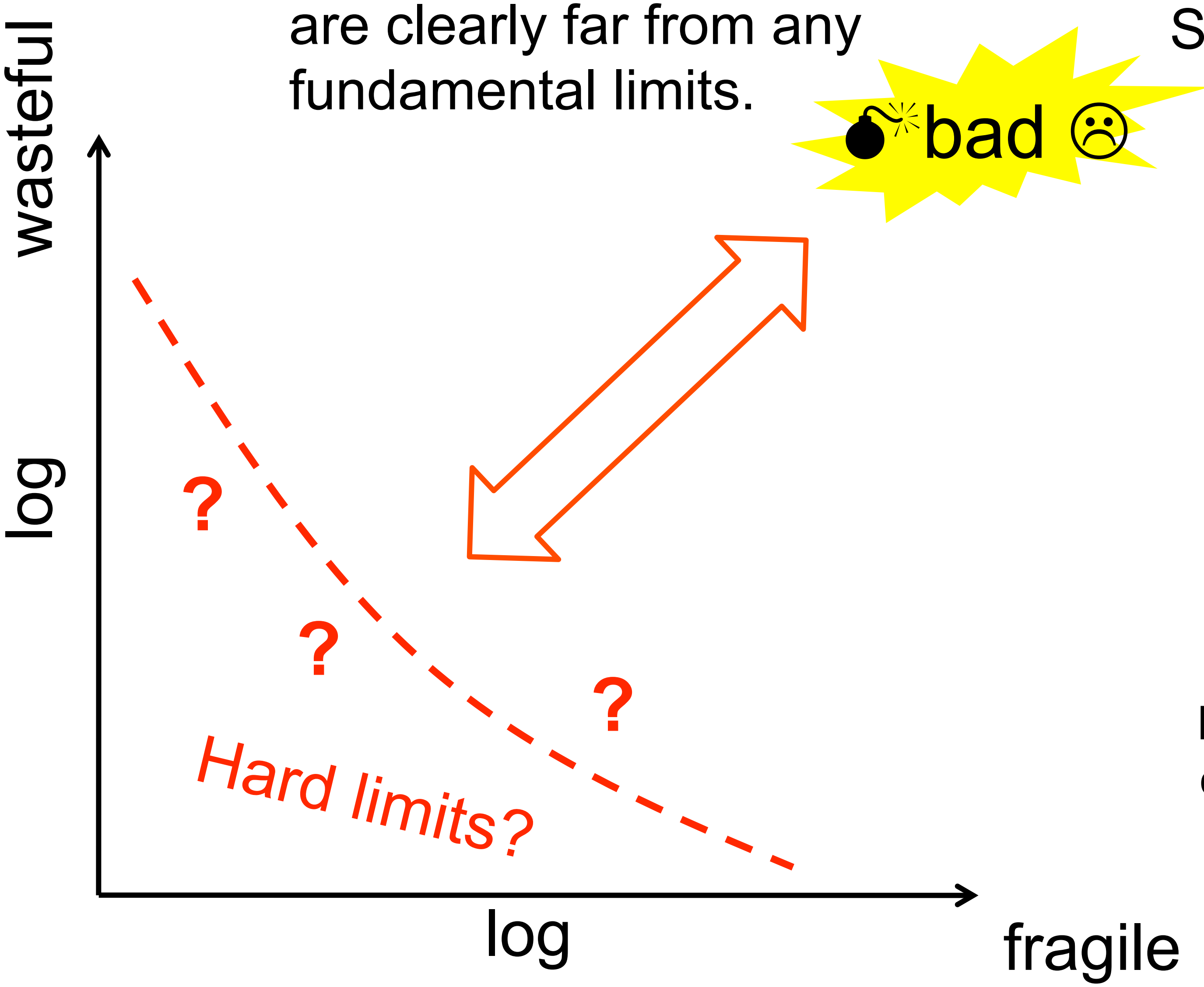
- Important tradeoffs are ***across*** these dimensions
- Speed vs efficiency vs robustness vs …
- Robustness is most important for complexity
- Collapse efficiency dimensions

wasteful



- Important tradeoffs are ***across*** these dimensions
- Speed vs efficiency vs robustness vs …
- Robustness is most important for complexity
- Collapse efficiency dimensions

But many existing systems and architectures are clearly far from any fundamental limits.



So fixing “bugs” in existing architectures has most immediate impact.

Note: “log” suggests orders of magnitude variations

Important Influences

- There increasingly many researchers/authors with increasingly coherent thinking about architecture
- Remarkably convergence across many fields
- Different language and domains so translation is difficult

New (and old) connections

- Biology/Medicine (Savageau, G&K, Mattick, Csete, Arkin, Alon, Caporale, de Duve, Exerc Physio, Acute Care, etc…)
- Internet (Kelly/Low, Willinger, Chang, Clark, Wroclawski, Alderson, Day, etc)
- Management (Baldwin,…)
- Resilience/Safety/Security Engineering/Economics (Wood, Anderson, …)
- Platform Based Design: Alberto S-V, Lee, …

Other Complex Influences

- Architecture (Alexander, Salingeros,…)
- Aerospace (many, Maier is a good book)
- Philosophy/History (Fox Keller, Jablonka&Lamb)
- Physics/ecology (Carlson)

First emphasis (+new)

- Internet (Day, Low, Willinger, Clark, Wroclawski,…)
- Statistical mechanics (Sandberg, Delvenne,…)
- Biology (lots…)

Other topics maybe later:

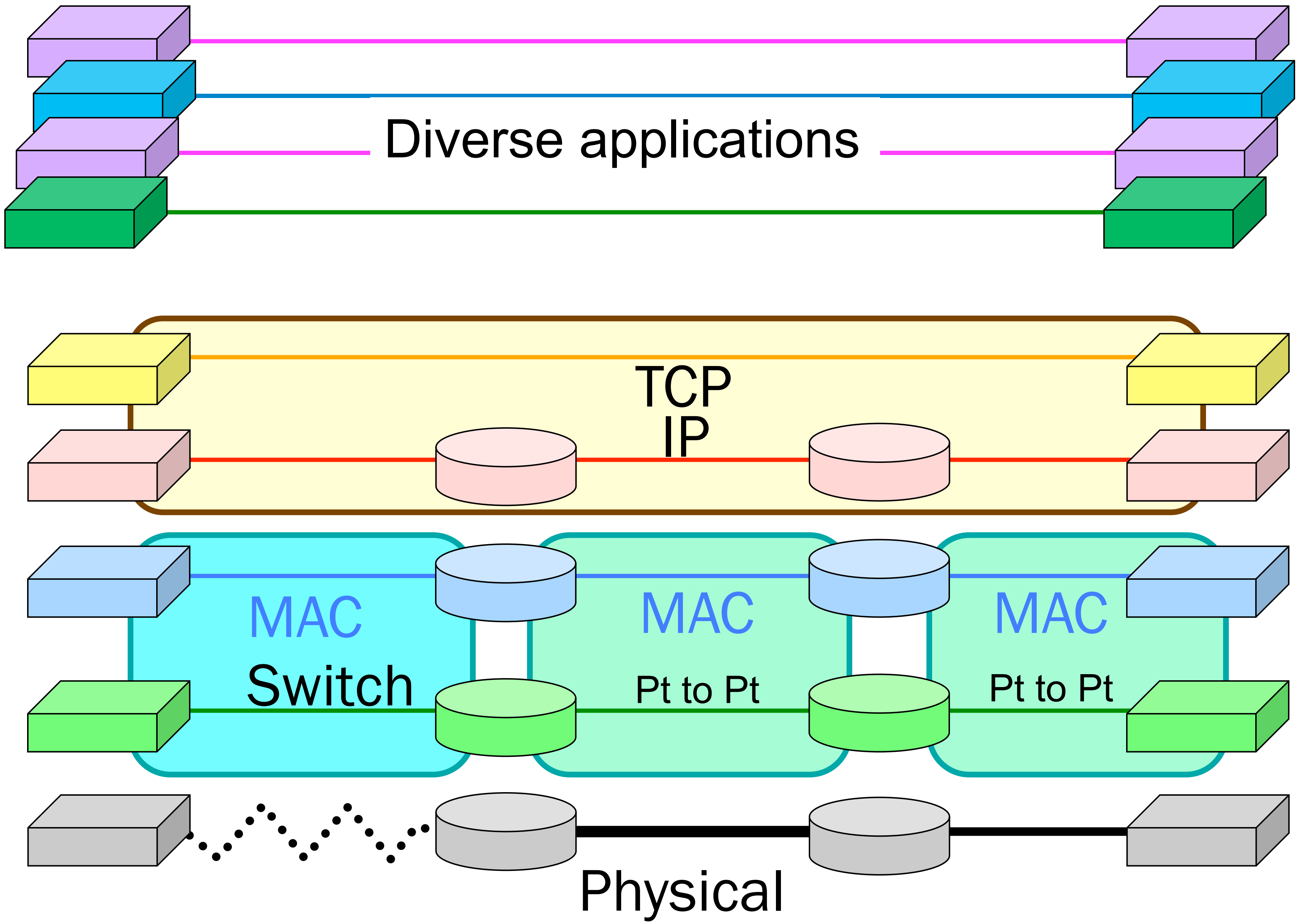
- Antennas and beam forming (Lavaei, Babakhani, Hajimiri)
- Shear flow turbulence (Gayme, McKeon, Bamieh)

Modern theory and the Internet

Levels of understanding	Topics
Verbal/cartoon	Traffic
Data and statistics	Topology
Modeling and simulation	Control and dynamics
Analysis	Layering
Synthesis	Architecture

Recent progress (1995-)

	Traffic	Topology	C&D	Layering	Architect.
Cartoon					?
Data/stat					
Mod/sim					
Analysis					
Synthesis					


Diverse applications
TCP
IP
MAC
Switch
MAC
Pt to Pt
MAC
Pt to Pt
Physical

Theoretical framework:
Constraints that deconstrain

Applications
Deconstrained

$$\min_{\mathbf{x}} \left\{ \int \left(\left\| R\tilde{\mathbf{x}} - \mathbf{c} \right\|^2 + \left\| R\mathbf{x} - \mathbf{c} \right\|^2 \right) dt \right\}$$

$$\left|\; \tilde{\mathbf{x}} = \arg\max_{\mathbf{v}} L(\mathbf{v}, \mathbf{p}), \quad \dot{\mathbf{p}} = R\mathbf{x} - \mathbf{c} \right.$$

$$\Rightarrow x_s = \arg\max_{v} L_s(v, \mathbf{p})$$

Resources
Deconstrained

Enormous progress
- Layering
- Optimization
- Optimal control
- Robust control
- Game theory
- Network coding

Continuing progress but clear limitations.

Theoretical framework:
Constraints that deconstrain

Enormous progress

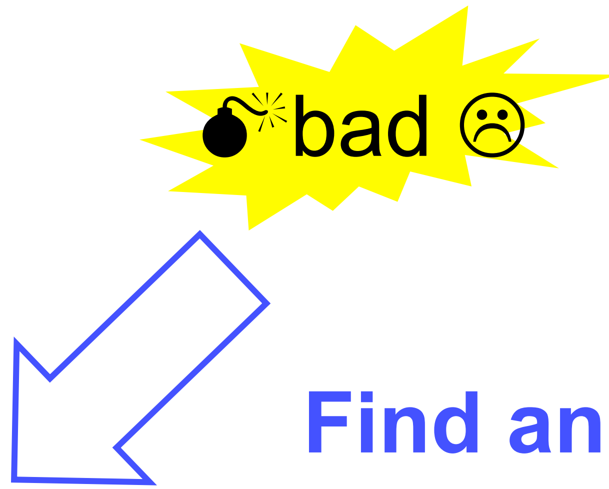
- Layering
- Optimization
- Optimal control
- Robust control
- Game theory
- Network coding

- Many robustness issues left unaddressed
- Secure, verifiable, manageable, maintainable, etc
- Architecture/policy, not part of control/dynamics
- How to expand the theory?
- What are obvious bugs?

- Note: Huge success of TCP/IP may have blinded us to historical artifacts, need theory-based rethinking bad ☹

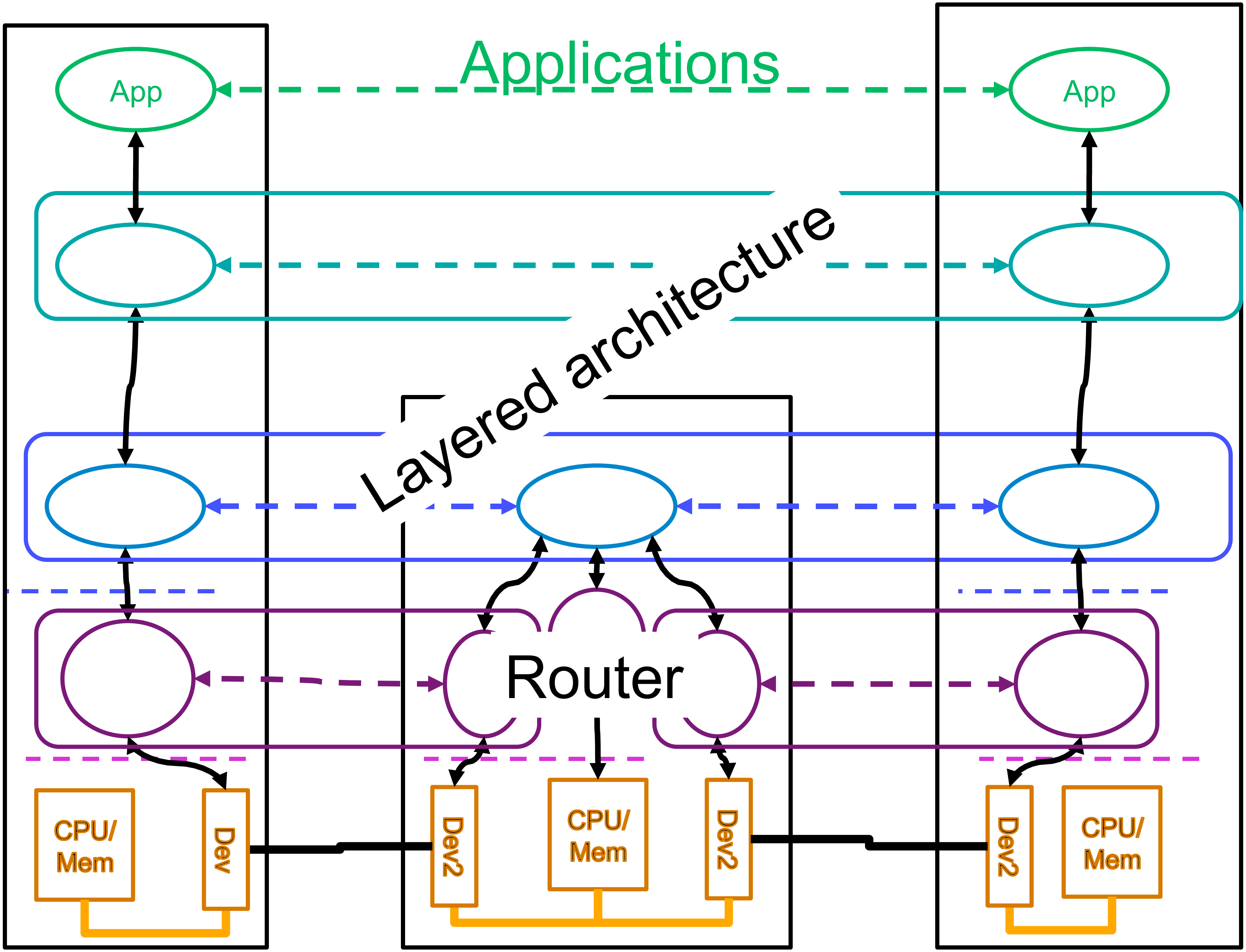
Find and fix bugs?

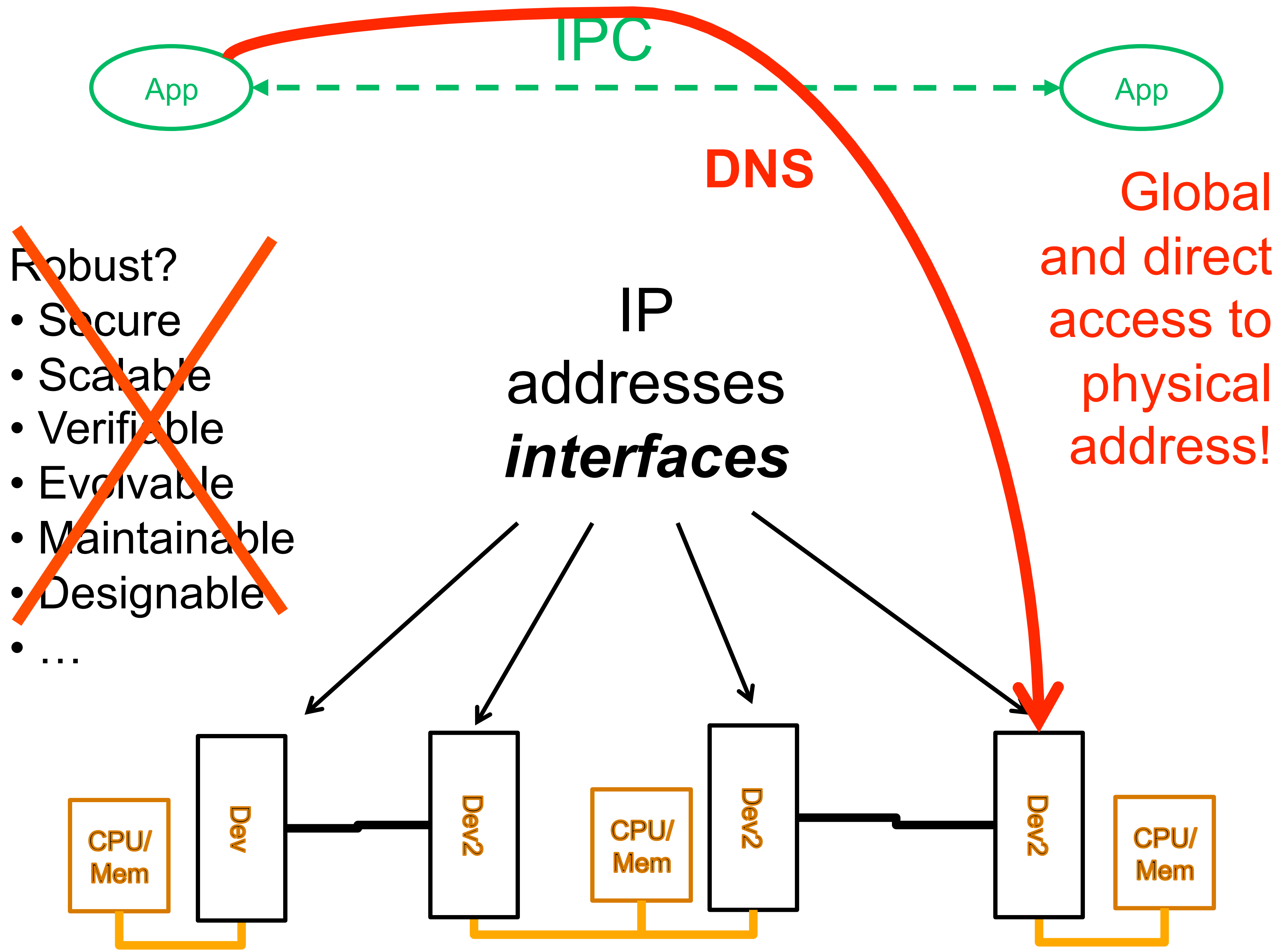
Patterns in Network Architecture (Day)



Find and fix bugs

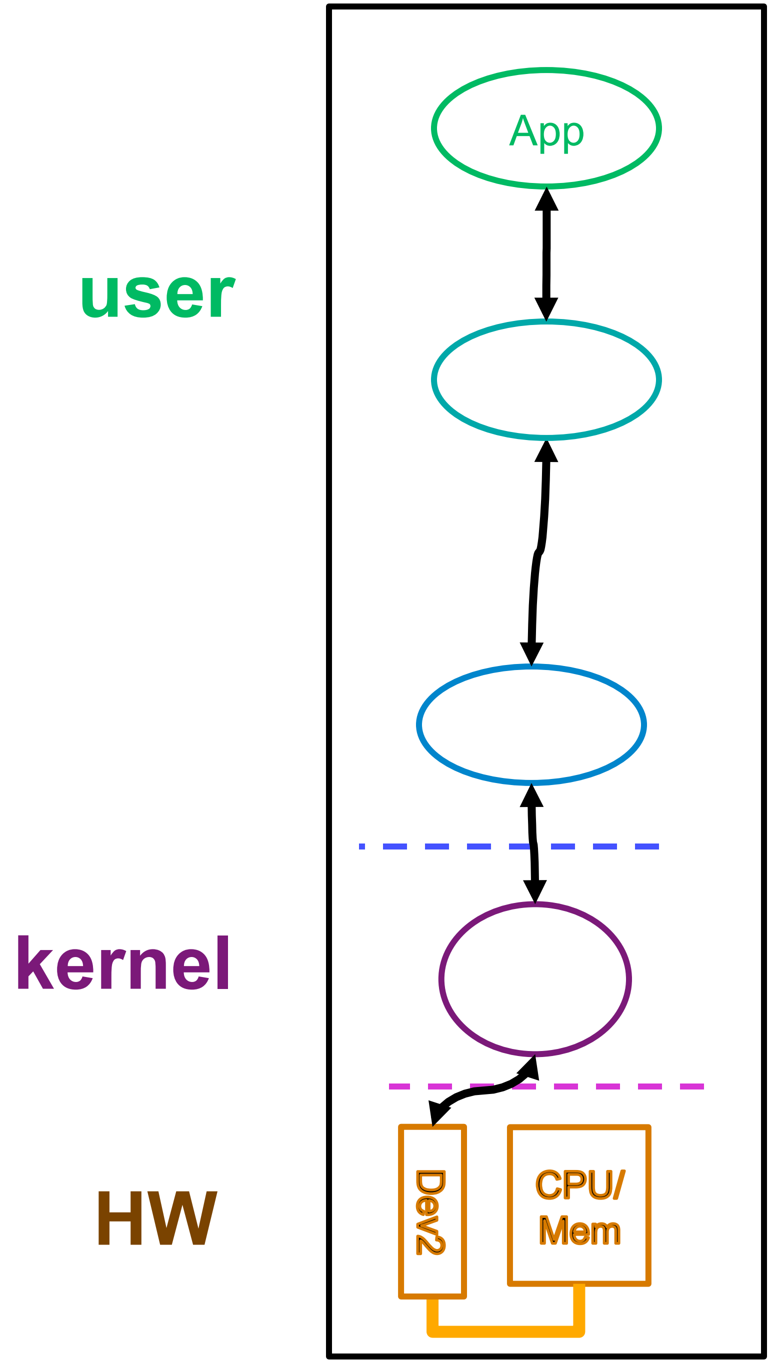
- Lots of well-known problems
- Emerging unifying framework
- PNA, RNA (Touch et al), etc
- Compatible with existing theory
- ***Illustrate with simple example***


Applications
App
App
Layered architecture
Router
CPU/
Mem
Dev
Dev2
CPU/
Mem
Dev2
Dev2
CPU/
Mem


IPC
App
App
DNS
Global and direct access to physical address!
Robust?
• Secure
• Scalable
• Verifiable
• Evolvable
• Maintainable
• Designable
• …
IP addresses *interfaces*
CPU/ Mem
Dev
Dev2
CPU/ Mem
Dev2
Dev2
CPU/ Mem

In operating systems: Don’t cross layers

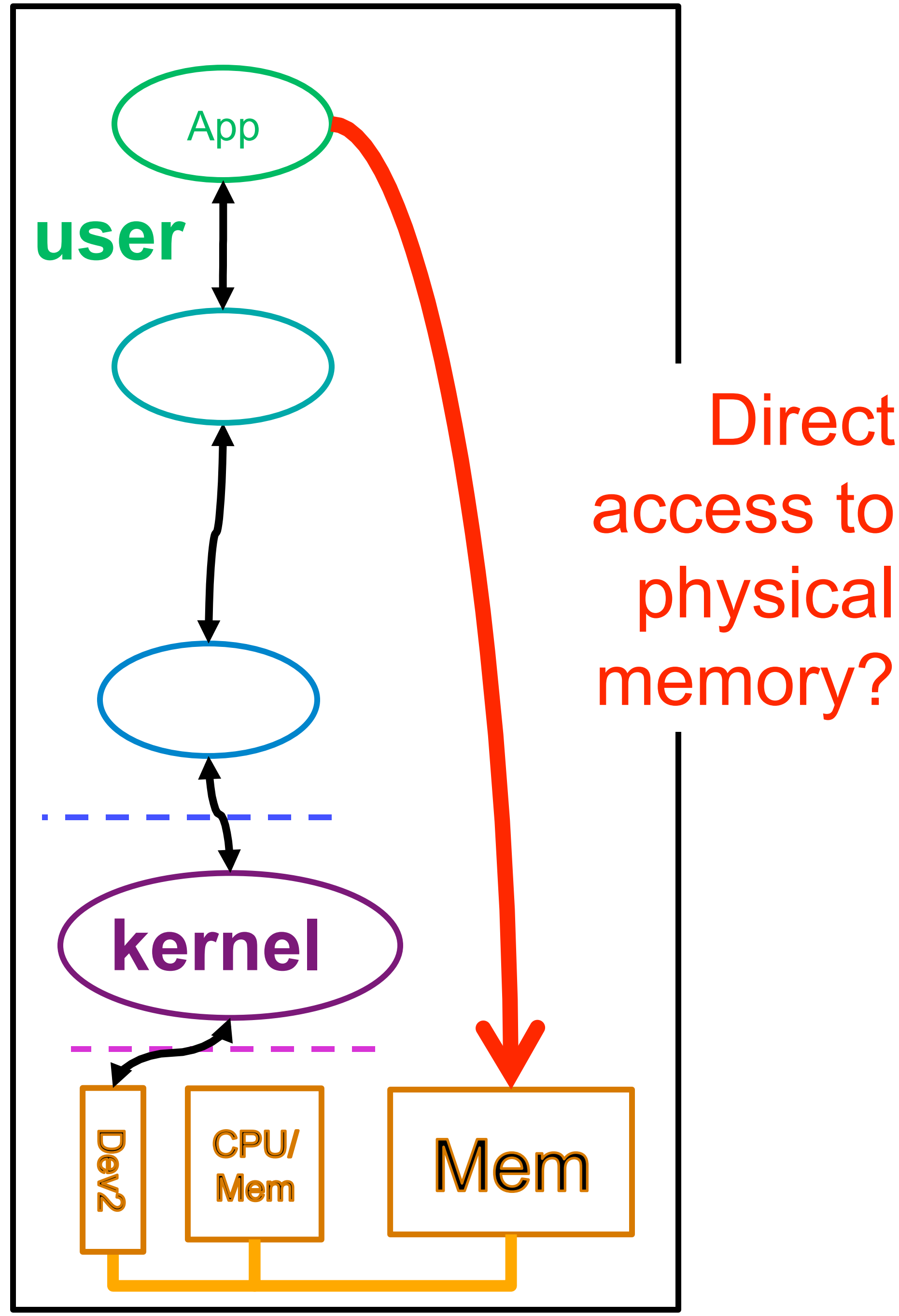
Simpler example



In operating systems: Don’t cross layers

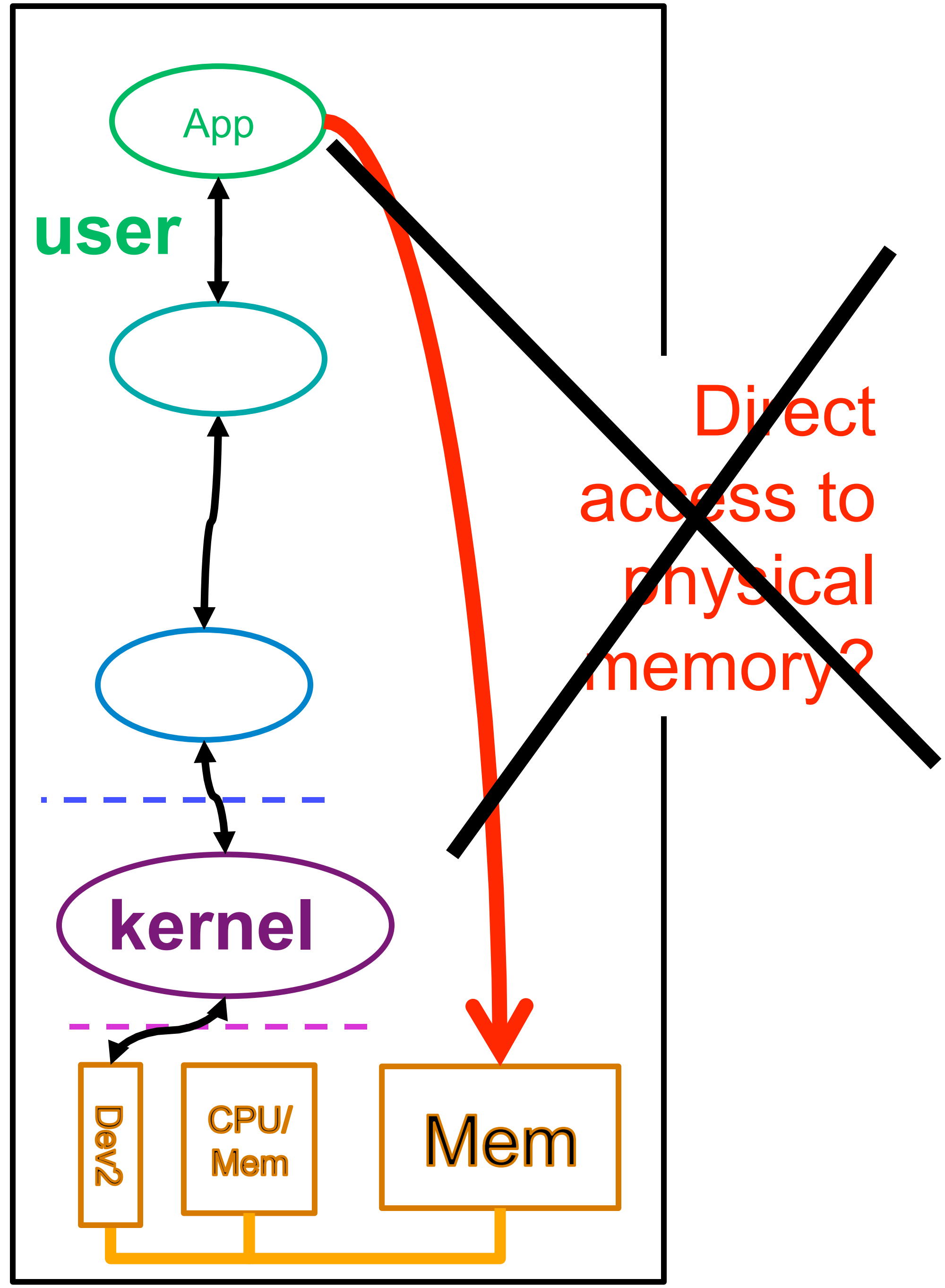
~~Robust~~

- ~~Secure~~
- ~~Scalable~~
- ~~Verifiable~~
- ~~Evolvable~~
- ~~Maintainable~~
- ~~Designable~~
- ~~…~~



In operating systems: Don’t cross layers

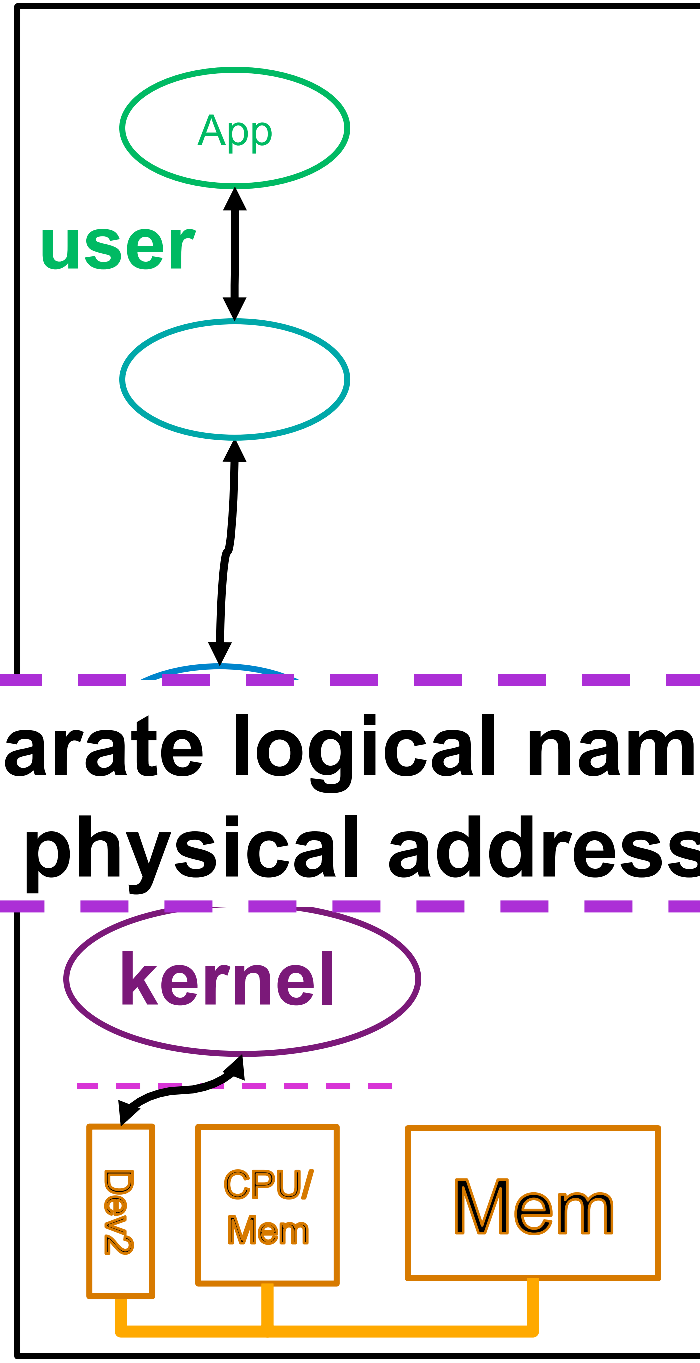
~~Robust~~
- ~~Secure~~
- ~~Scalable~~
- ~~Verifiable~~
- ~~Evolvable~~
- ~~Maintainable~~
- ~~Designable~~
- ~~…~~



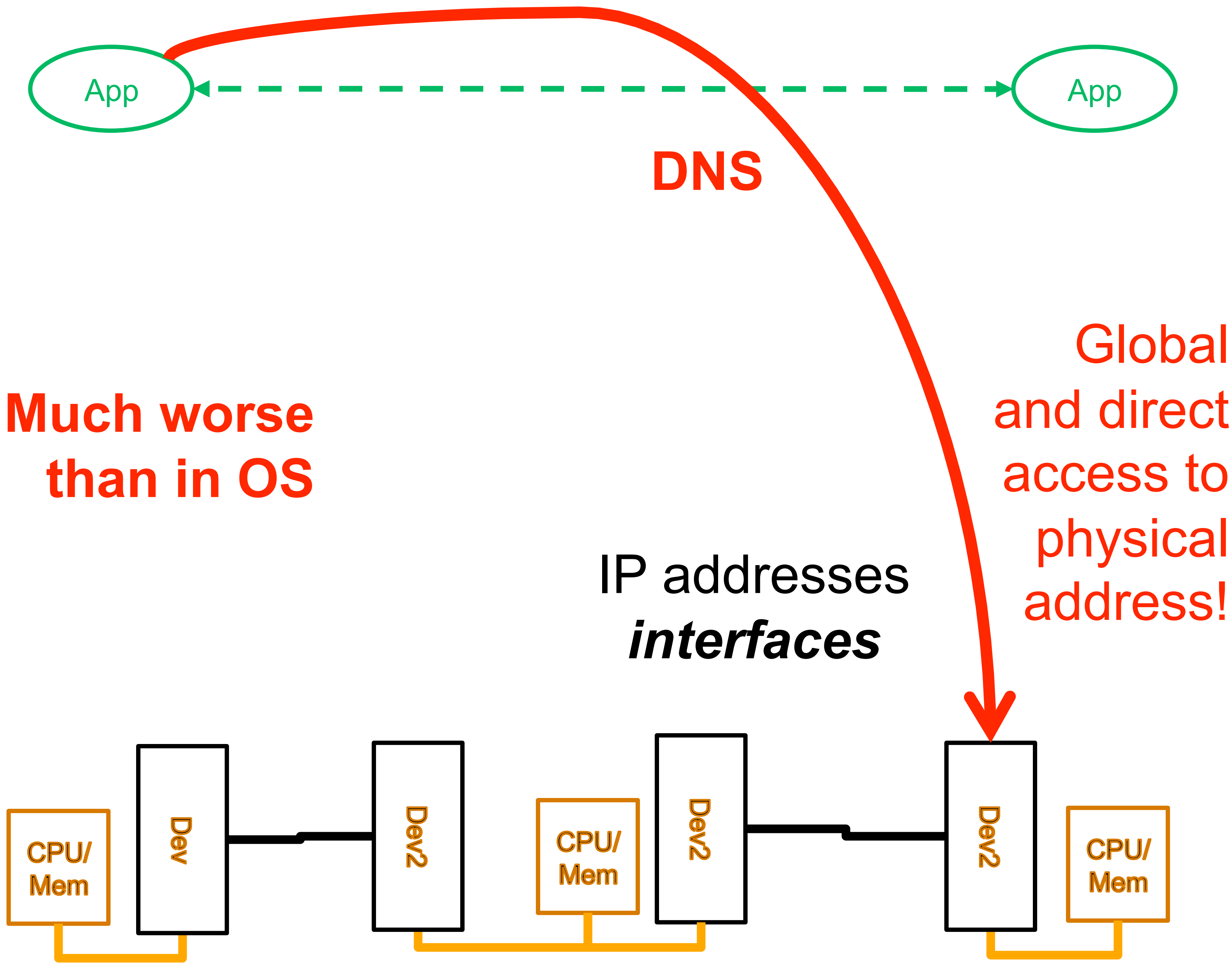
In operating systems: Don't cross layers (E2E)

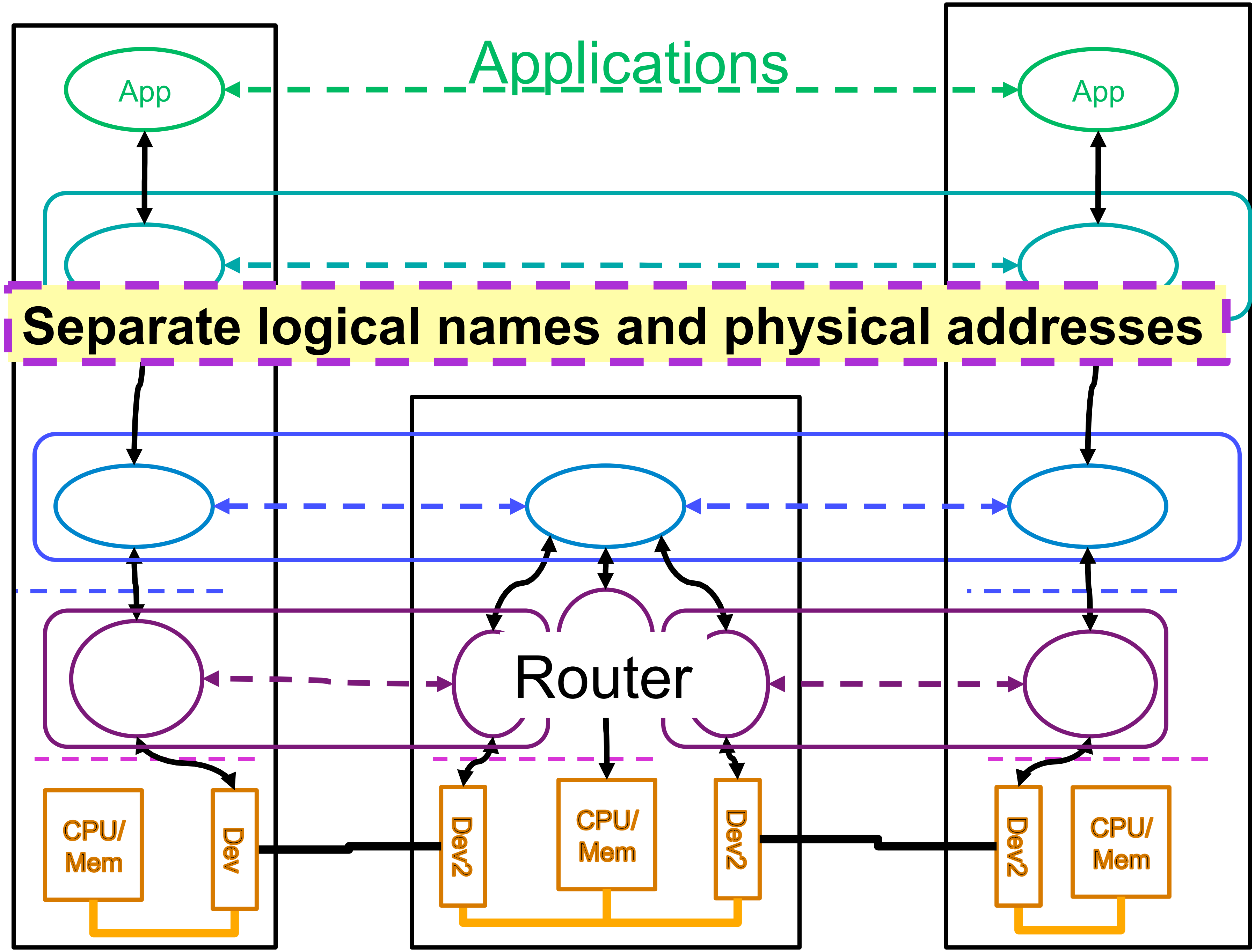
Robust

- Secure
- Scalable
- Verifiable
- Evolvable
- Maintainable
- Designable
- …



Separate logical names and physical addresses


App
App
DNS
Much worse than in OS
Global and direct access to physical address!
IP addresses *interfaces*
CPU/ Mem
Dev
Dev2
CPU/ Mem
Dev2
Dev2
CPU/ Mem



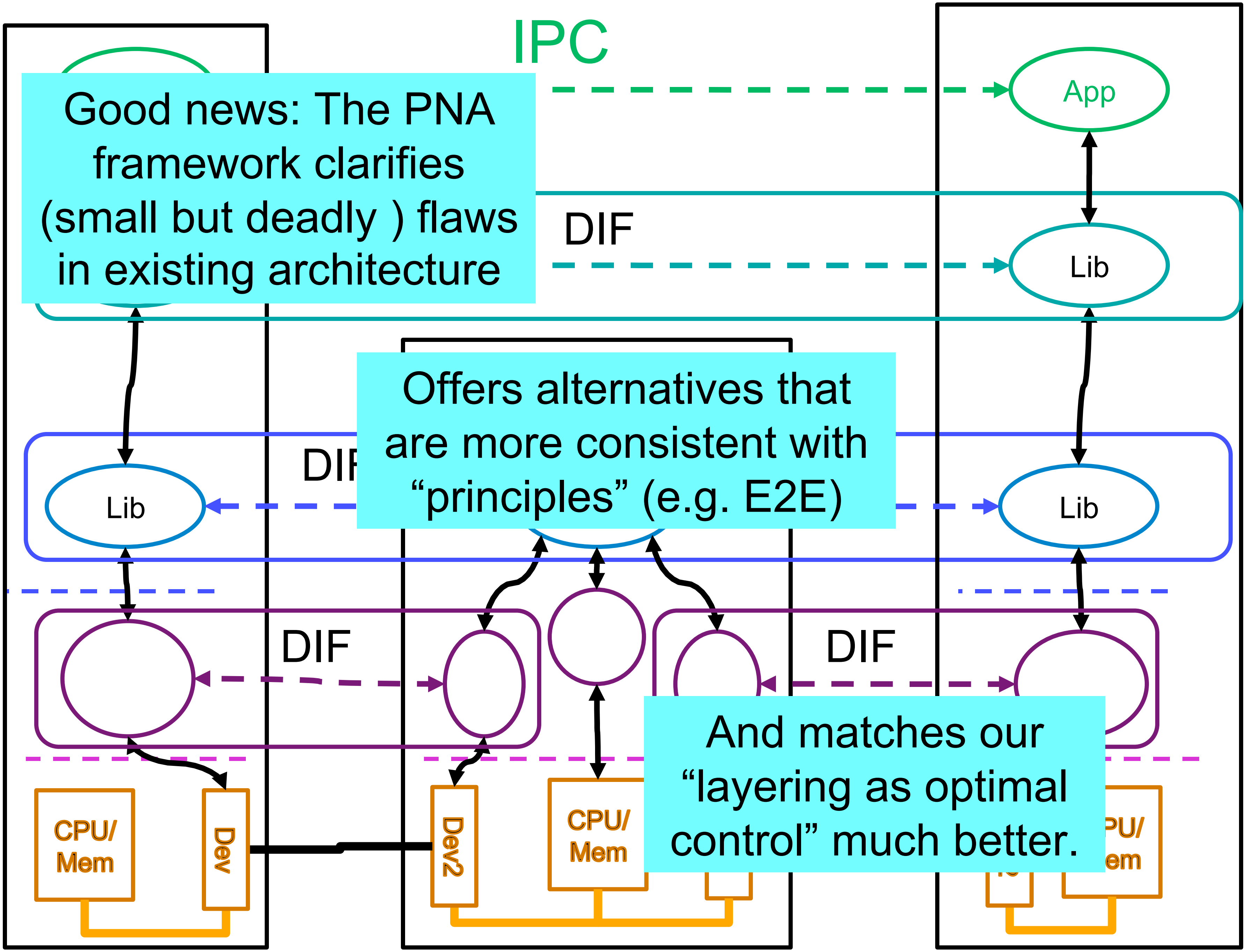
Applications
App
App
Separate logical names and physical addresses
Router
CPU/ Mem
Dev
Dev2
CPU/ Mem
Dev2
Dev2
CPU/ Mem

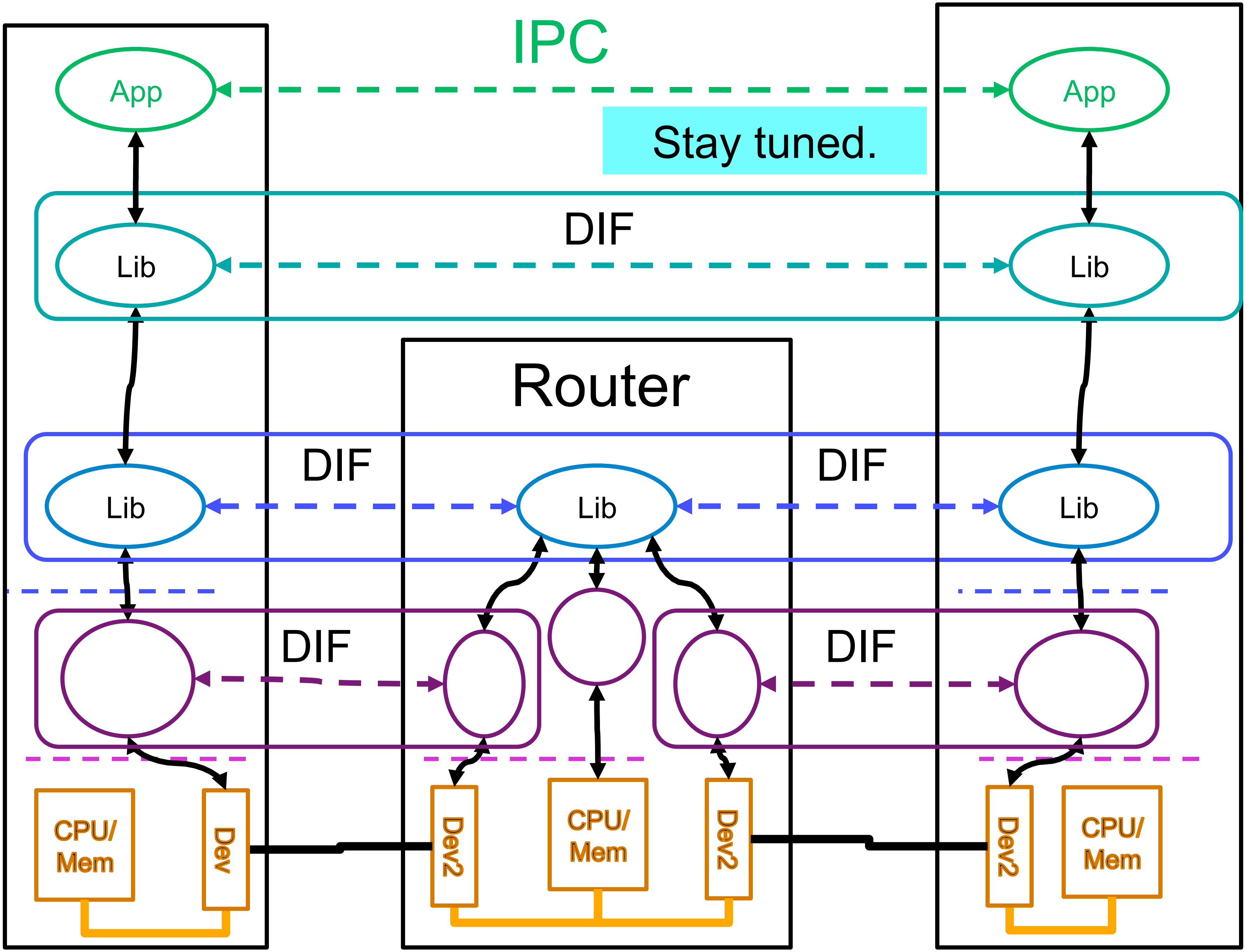
Other insecurities in TCP/IP?

Well known attacks

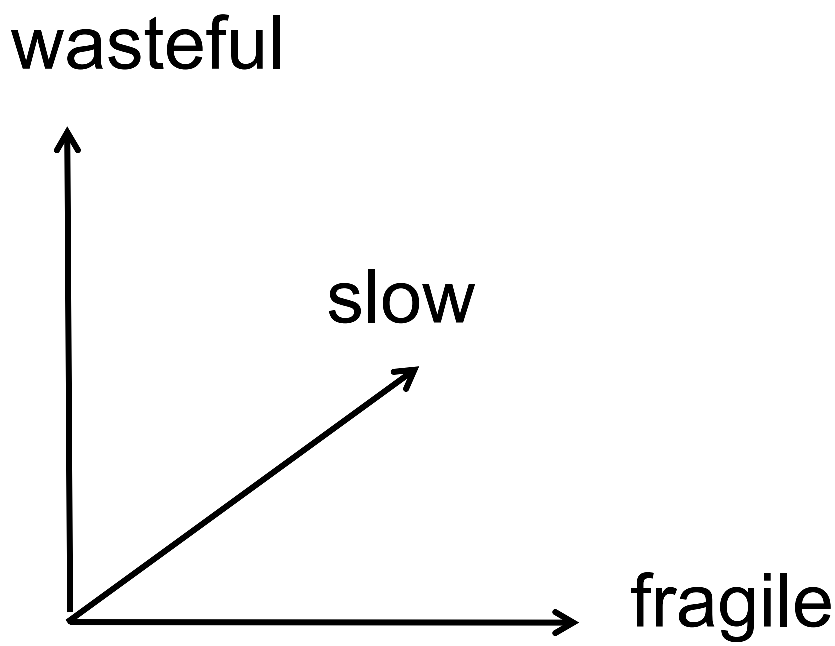
- port-scanning (why “well-known ports”?)
- connection-opening
- data-transfer
- Etc etc

- These are hard to fix in existing architecture
- Good news is alternatives may be easier than we think


IPC
App
Good news: The PNA framework clarifies (small but deadly) flaws in existing architecture
DIF
Lib
Offers alternatives that are more consistent with “principles” (e.g. E2E)
Lib
Lib
DIF
DIF
And matches our “layering as optimal control” much better.
CPU/ Mem
Dev
Dev2
CPU/ Mem


IPC
App
App
Stay tuned.
DIF
Lib
Lib
Router
DIF
DIF
Lib
Lib
Lib
DIF
DIF
CPU/
Mem
Dev
Dev2
CPU/
Mem
Dev2
Dev2
CPU/
Mem

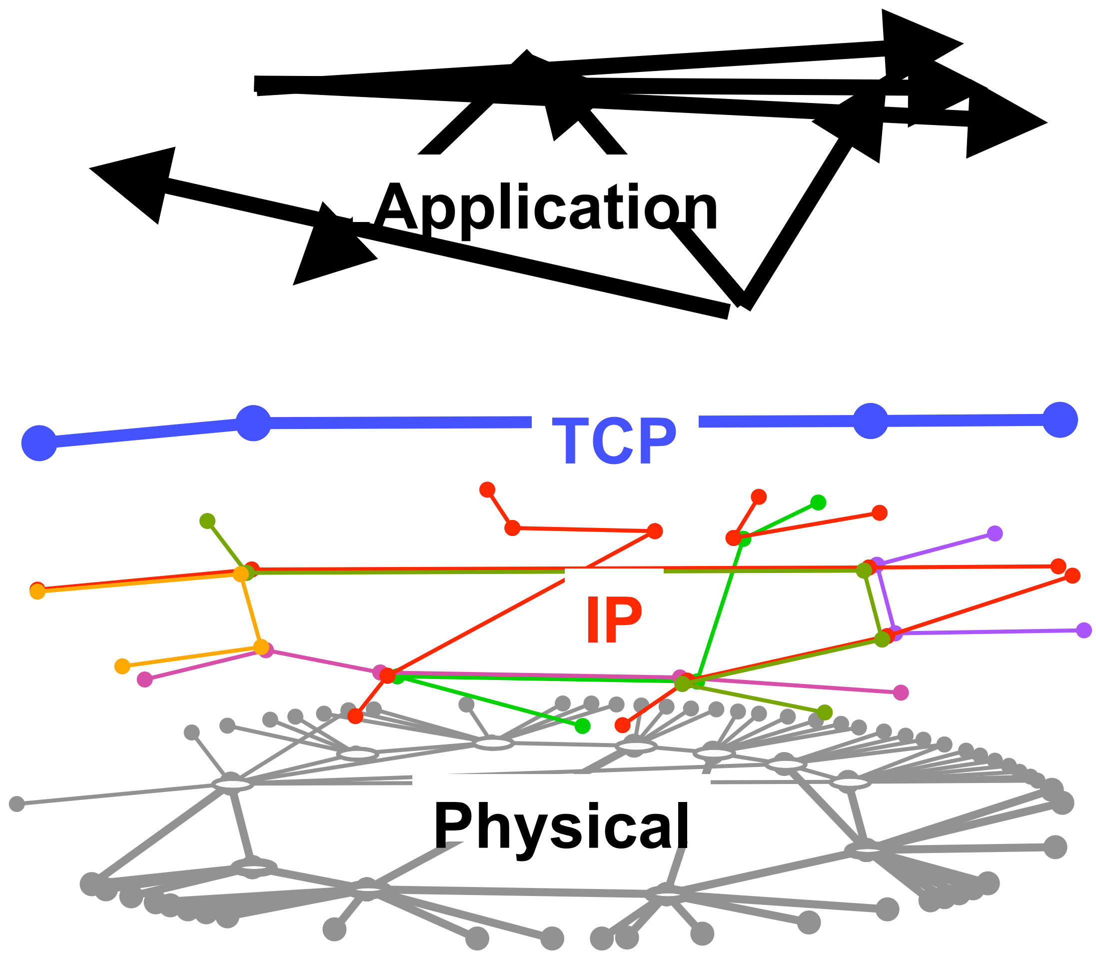
Naming and addressing
- need to match their layer
- translate/resolve between layers
- not be exposed outside layer
- ***familiar tradeoffs here***



Tradeoffs
- Addressing complexity
- Table sizes
- Forwarding
- Optimal routes
- Table updates

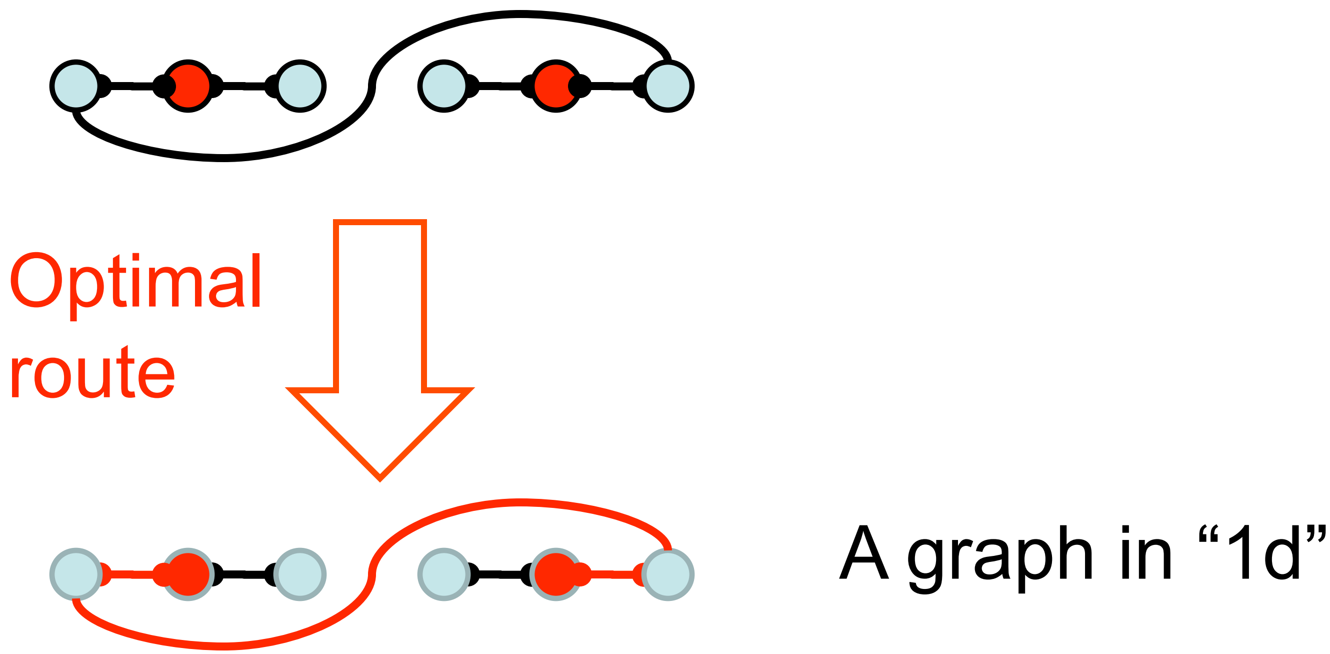
Naming and addressing
• need to match their layer
• translate/resolve between layers
• not be exposed outside layer

Architecture issues
• DNS
• NATS
• Firewalls
• Multihoming
• Mobility
• Routing table size
• Overlays
• …

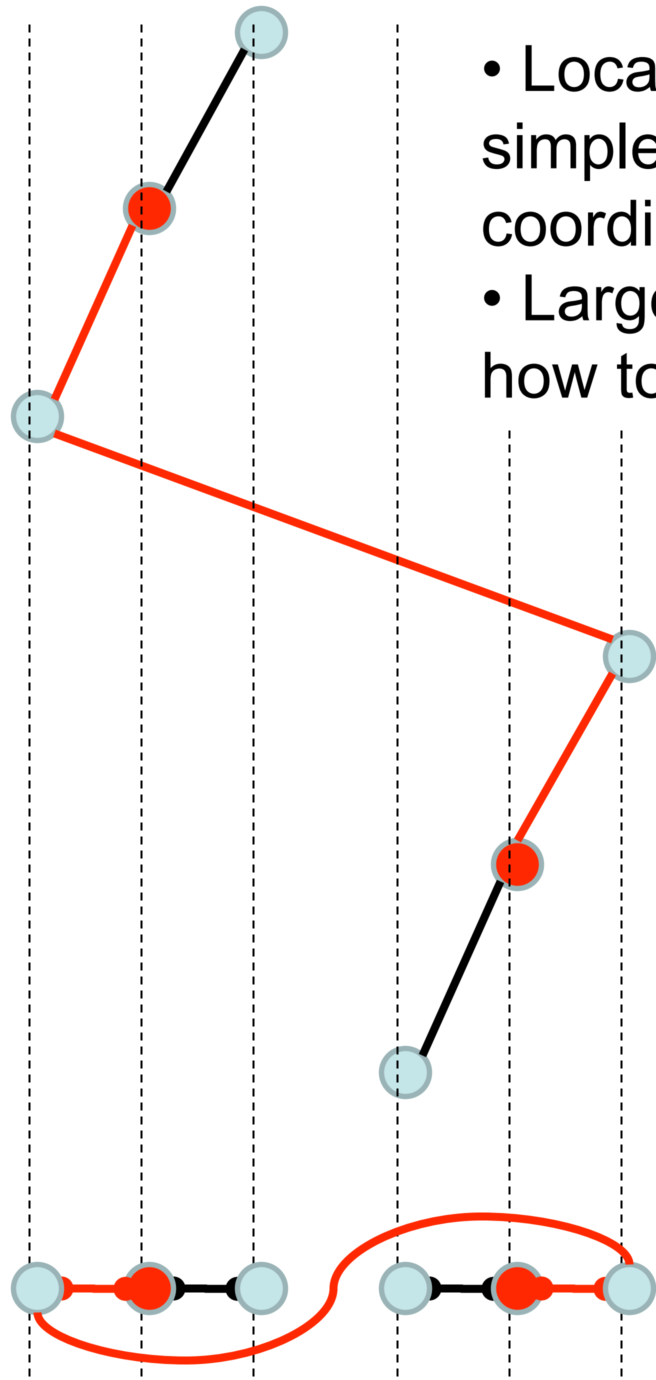


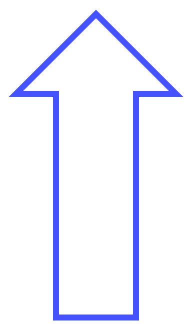
Trivial toy example

Consider a 1 dimensional geography
- Assume some link connectivity
- Optimal route might be indirect
- Consider route between red nodes

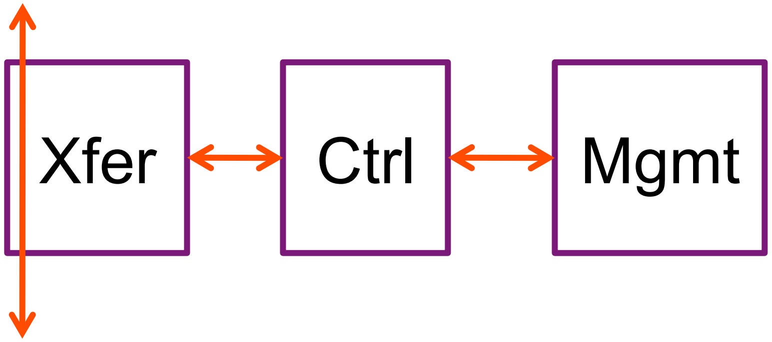


• Local, greedy routing using simple norms and “virtual coordinates” is globally optimal
• Large and growing literature on how to do this systematically



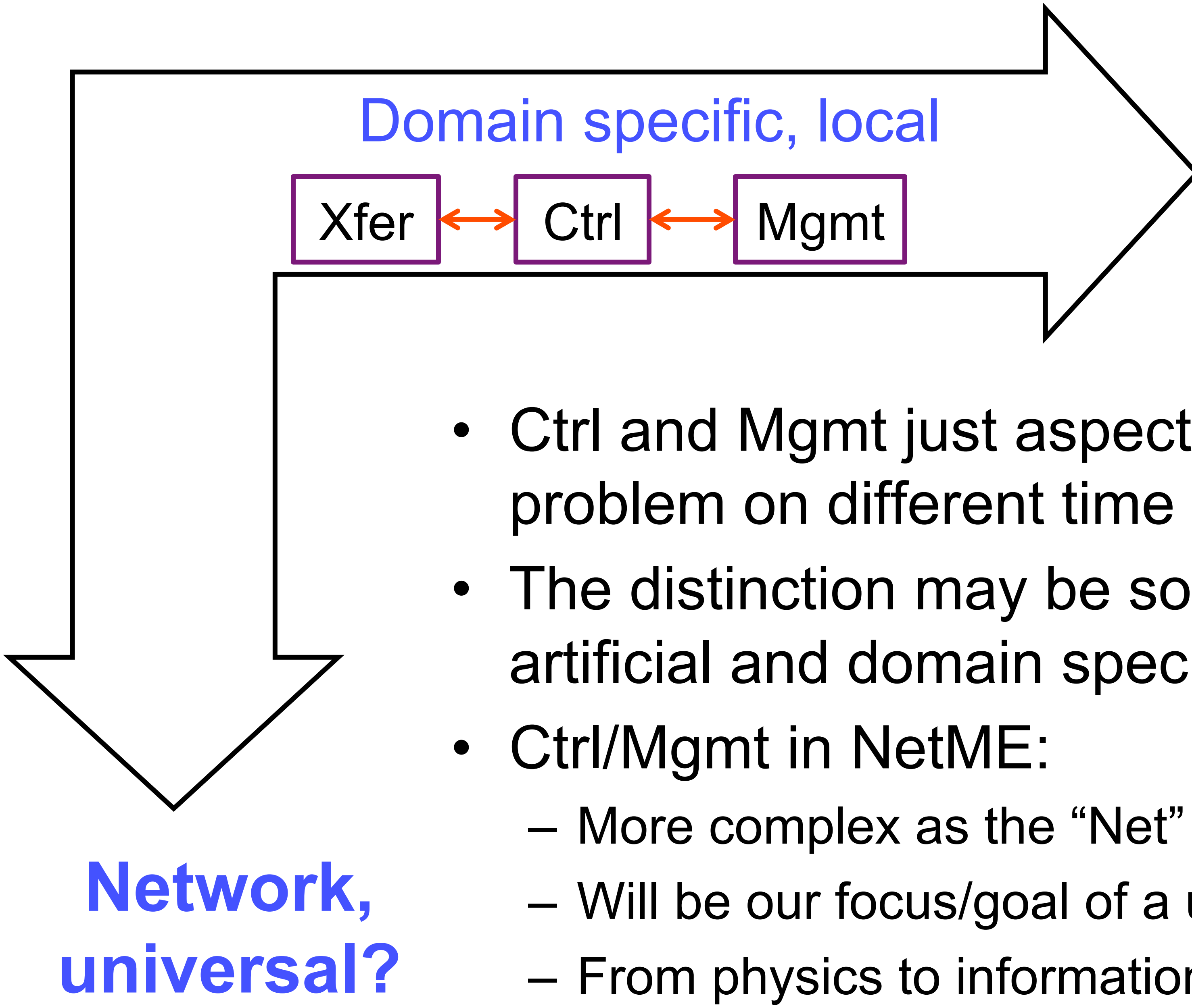


Add a virtual dimension



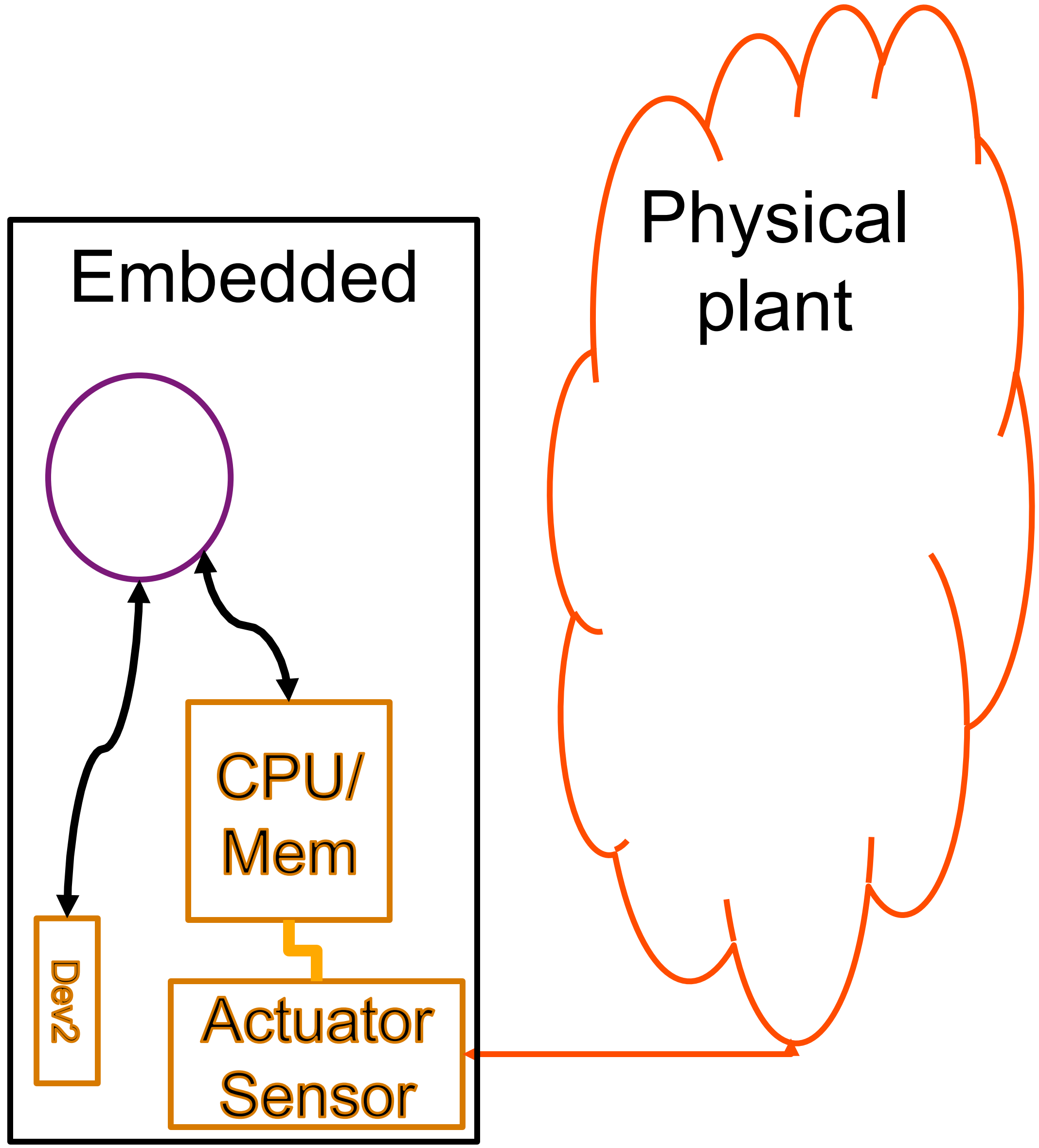
Universal functions?

- Transfer or transform (fastest)
  - Domain specific (data, power, goods, etc)
  - Depends on demand and supply of resources
- Control (middle)
  - Schedule/MUX resources in time and space
  - Flow and error control
- Management (slowest)
  - ***What*** resources are available?
  - ***Where*** are they?
  - Cost? Risk? etc

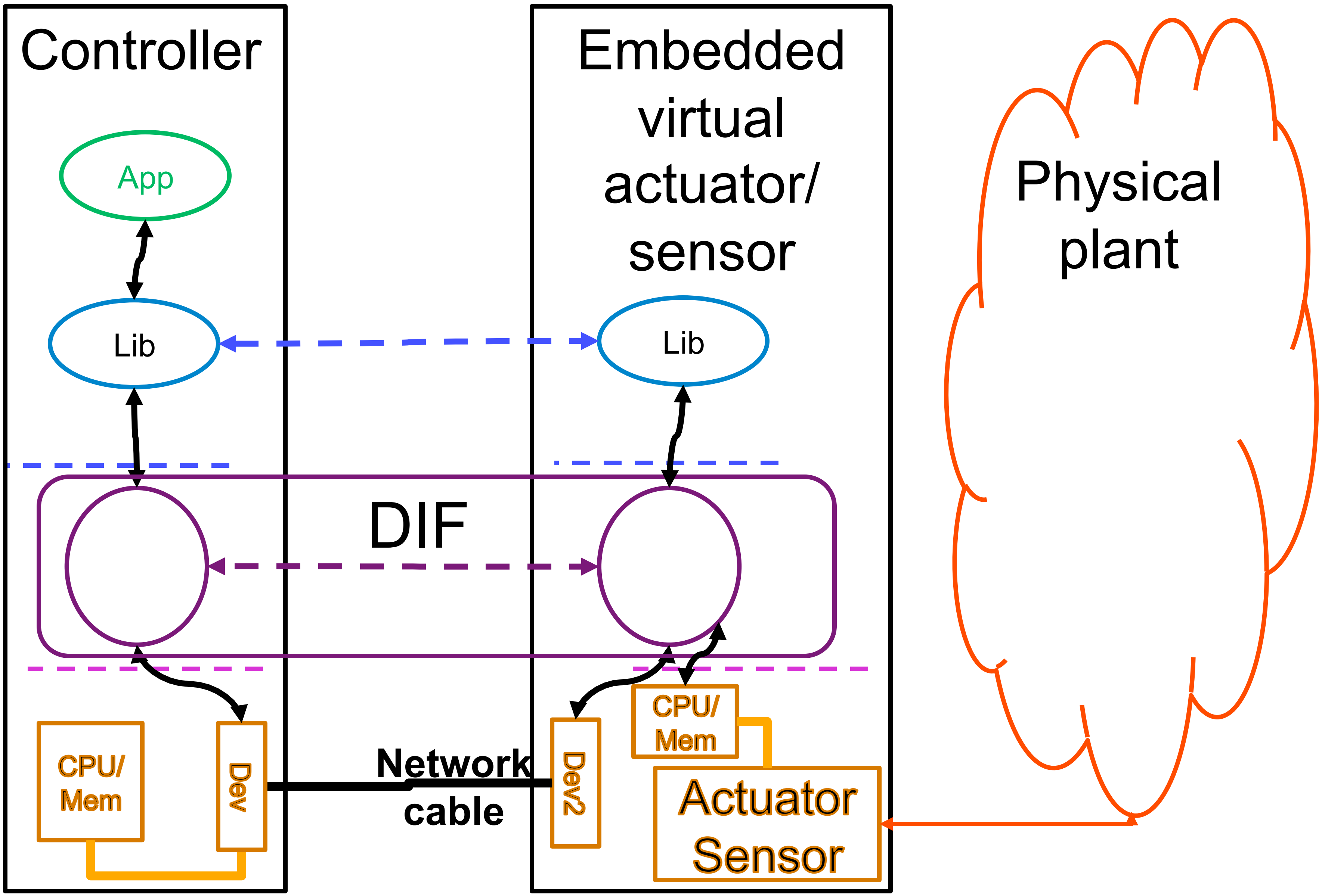


- Ctrl and Mgmt just aspects of a single problem on different time scales
- The distinction may be somewhat artificial and domain specific
- Ctrl/Mgmt in NetME:
  - More complex as the “Net” part grows
  - Will be our focus/goal of a unified theory
  - From physics to information to computation to control

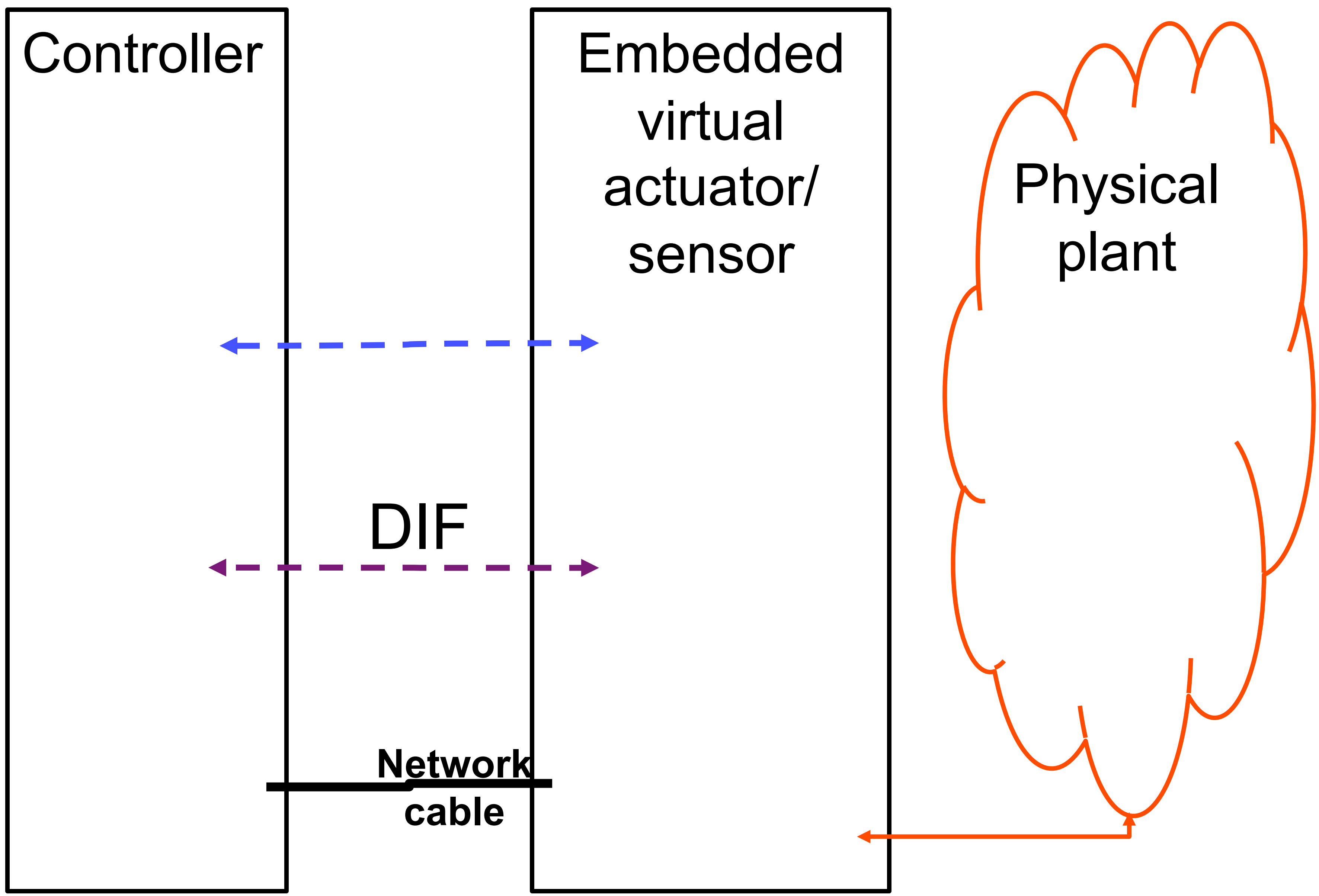
Embedded



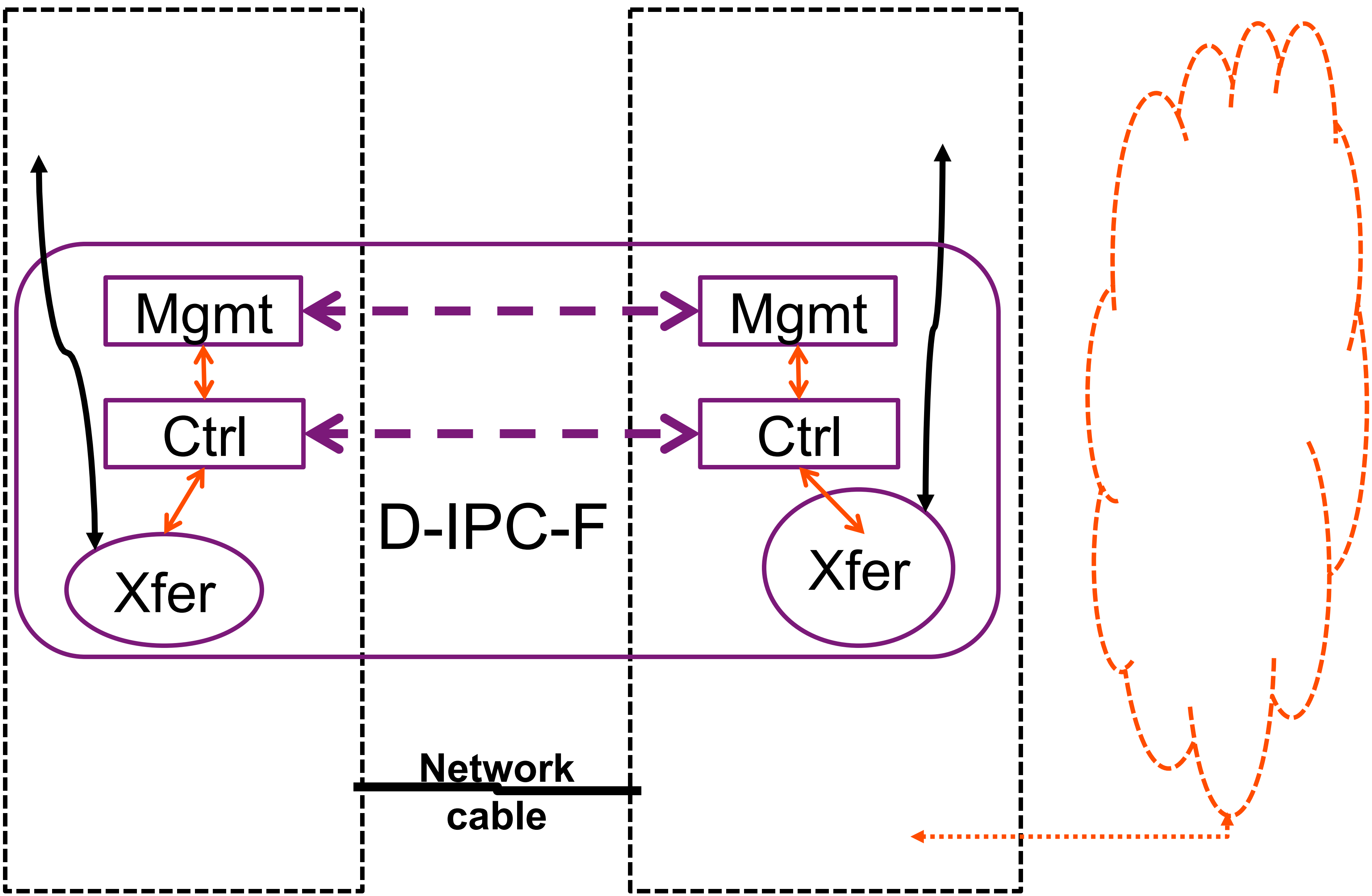
Networked embedded



Meta-layering of cyber-phys control

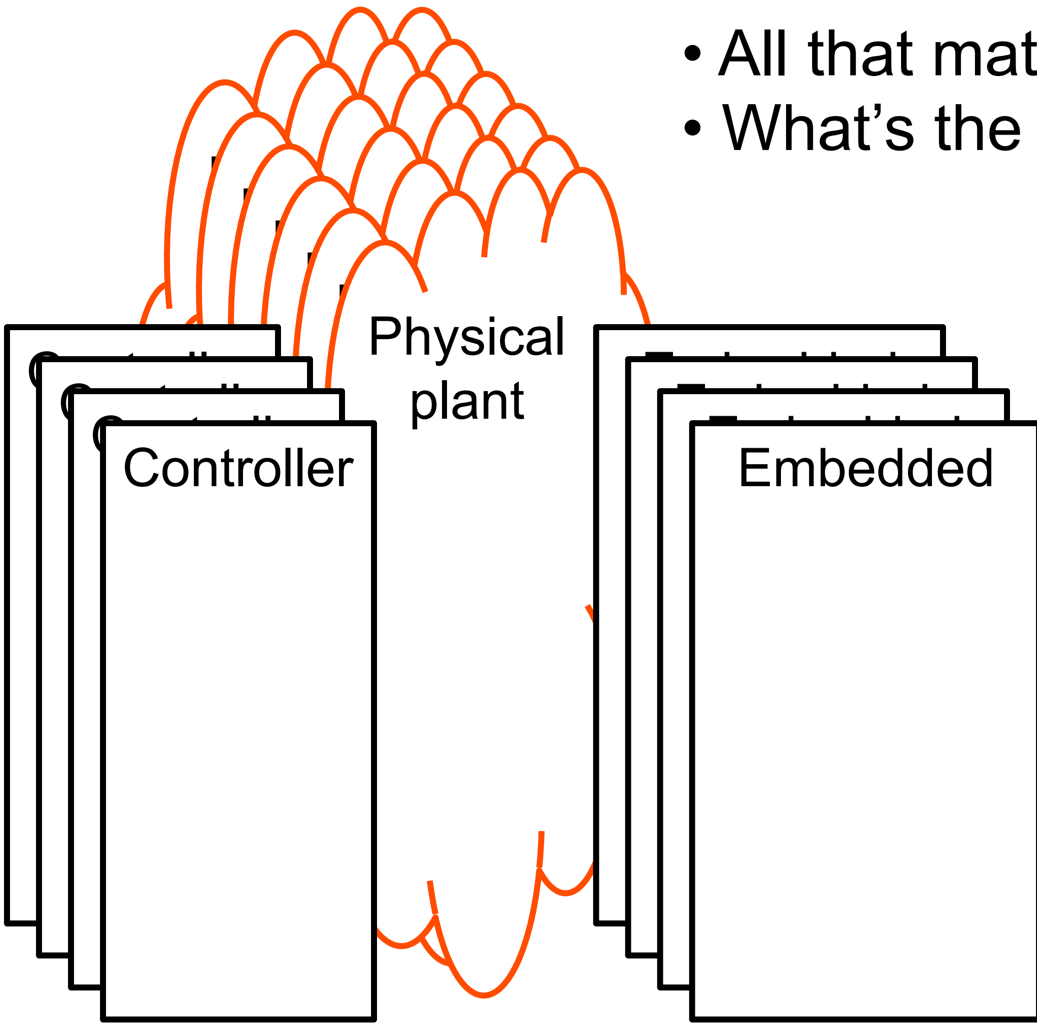


Micro-layering of D-IPC-F

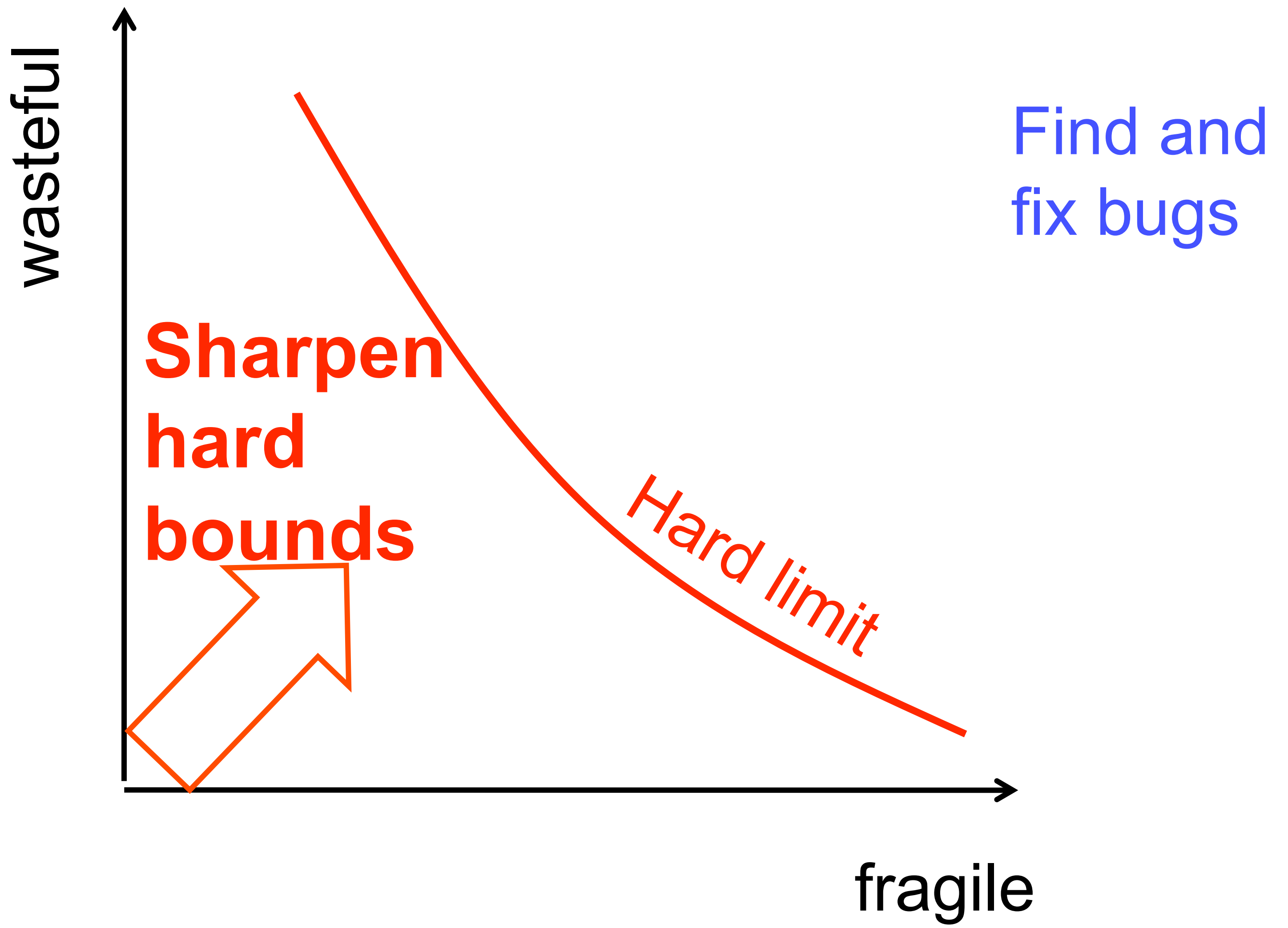


Smartgrid and cyberphys

- Everything is networked.
- Flows of data and power.
- All that matters is ***action***.
- What's the right architecture?



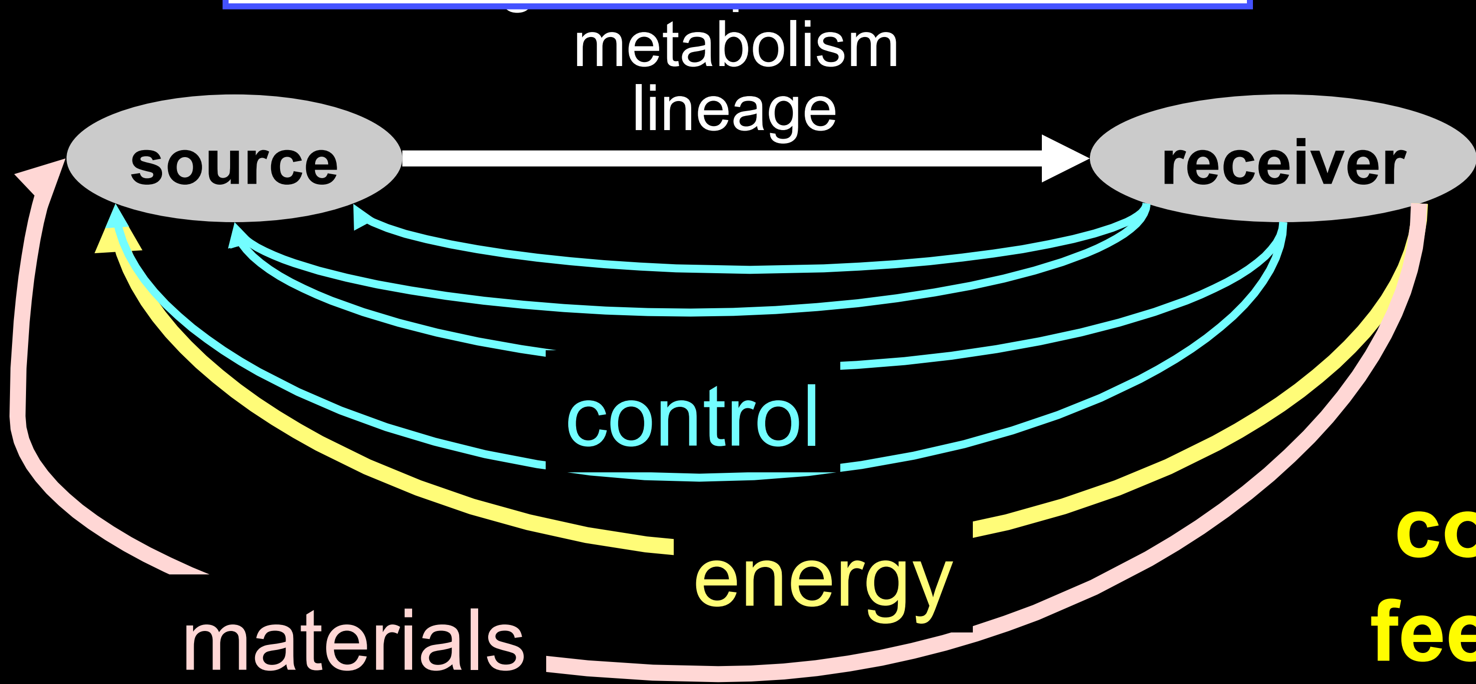
Complementary approaches



Previously: Hard tradeoffs between control and metabolic efficiency

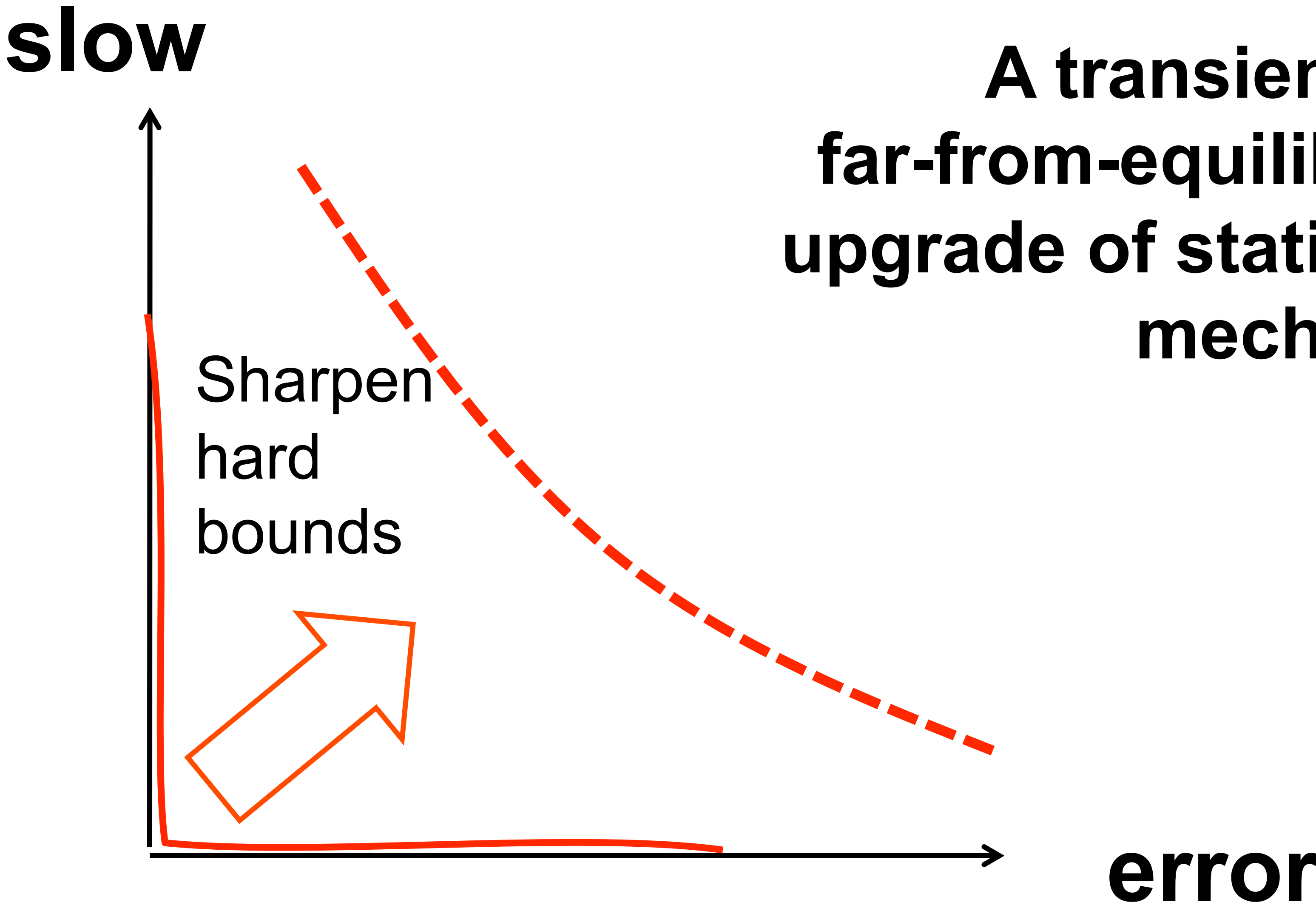
New: more mechanistic details for "efficiency" in control context

$$\frac{1}{\pi}\int_0^\infty \ln\left|S(j\omega)\right| \frac{z}{z^2+\omega^2}\,d\omega \ge \ln\left|\frac{z+p}{z-p}\right|$$



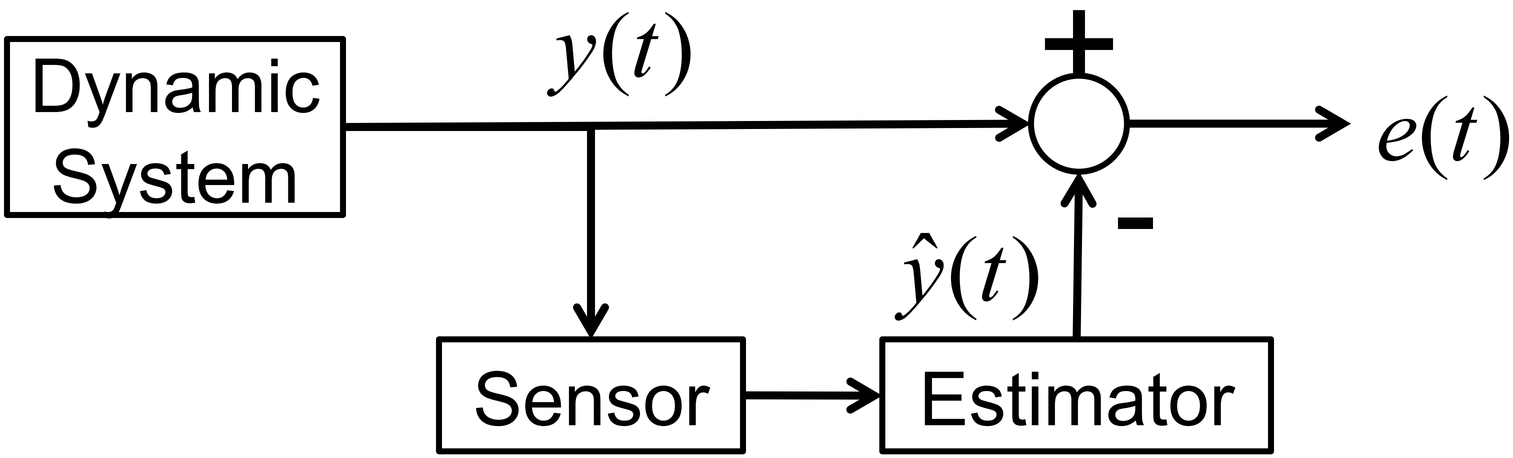
More complex feedback

Example



A transient and far-from-equilibrium upgrade of statistical mechanics

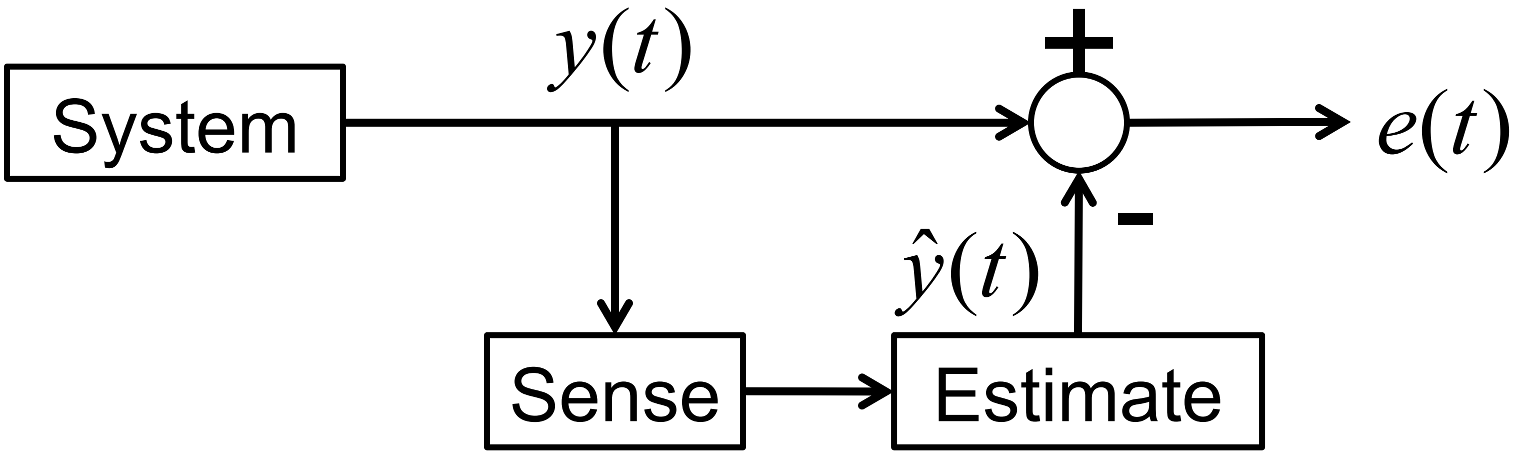
Estimation



Ideally $E\left(e^2(t)\right) \approx 0$

Estimation

Ideally $E\left(e^2(t)\right) \approx 0$



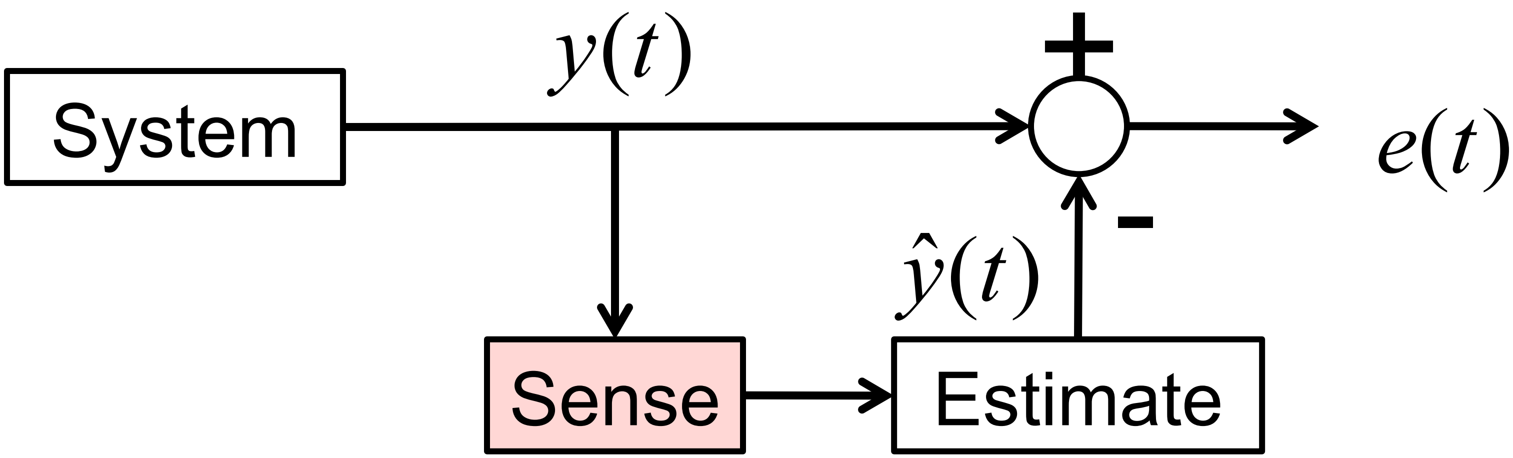
Realistically:

- Sensor "noise"
- Back action

Why? What are the consequences?

- Focus on fast transients (speed versus error).
- Asymptotic equilibrium same as standard physics.

Sensor "noise" assumptions

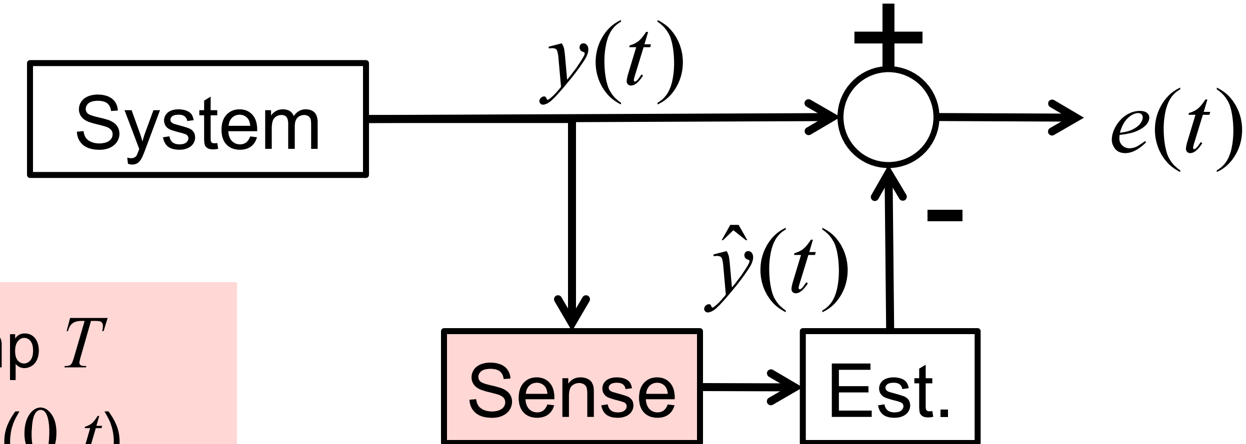


Sensor physics (phenomenology):

- Micro: many degrees of freedom
- Micro: energy conserving
- ⇒Macro: heat kT at temperature T, and noise

Use CDS tools to rigorously capture transients and nonequilibrium dynamics

Sensor "noise"



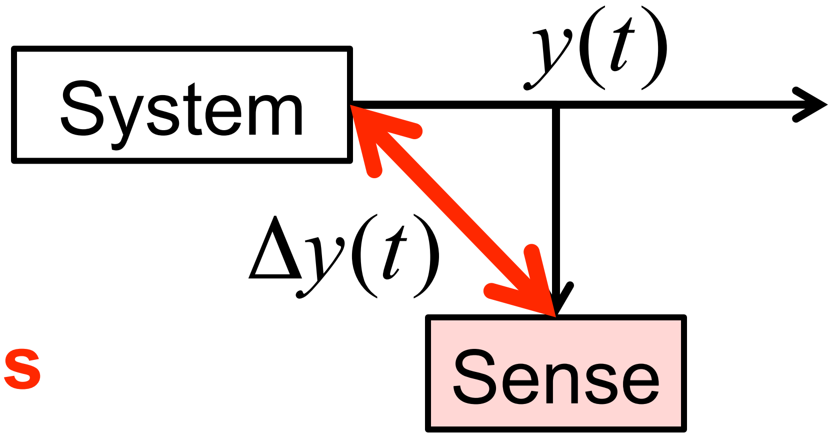
- Sensor at temp T
- Short interval $(0,t)$

Optimal estimator has hard limit

$$E\left(e^2(t)\right) \propto \frac{kT}{t} + O(1)$$

- Boltzmann constant k
- Units-dependent constants not shown, important in practice
- Asymptotic equilibrium recovers standard stat mech theory

Back action assumptions

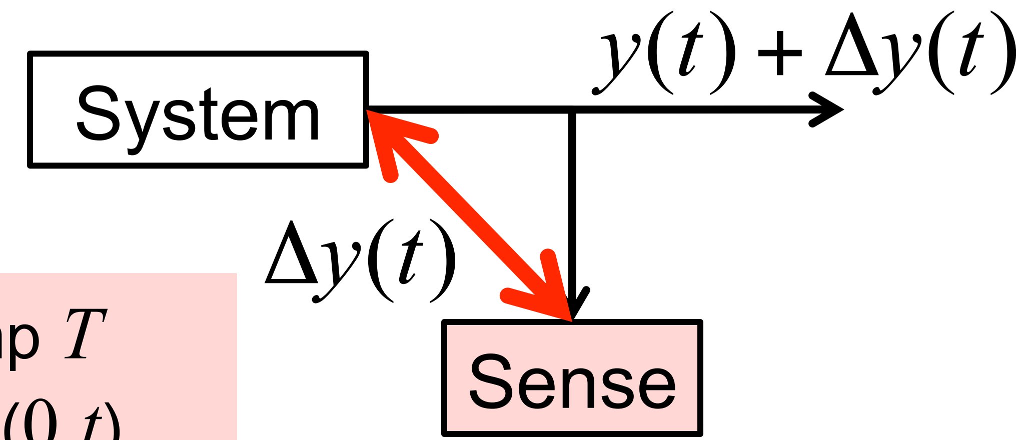


Sensor physics (phenomenology):

- Micro: entire system energy conserving
- $\Rightarrow$Macro: nontrivial impedance in sensor
- Note: this occurs even classically, and even if sensor has infinite energy supply (assume for now)

Use CDS tools to rigorously capture transients and nonequilibrium dynamics

Back action

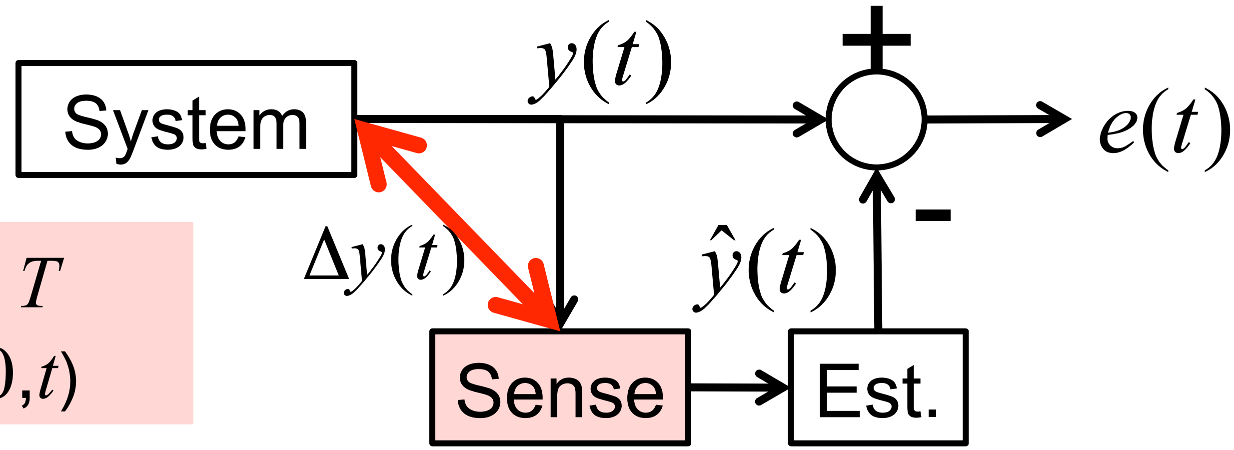


- Sensor at temp T
- Short interval $(0,t)$

Minimal back action has hard limit

$$E\left(\Delta y^2(t)\right) \propto kTt + O\left(t^2\right)$$

- Units-dependent constants not shown, important in practice
- Asymptotic equilibrium recovers standard stat mech theory



- Sensor at temp T
- Short interval $(0,t)$

$$E\left(\Delta y^2(t)\right) \geq kTt + O\left(t^2\right)$$

Back action

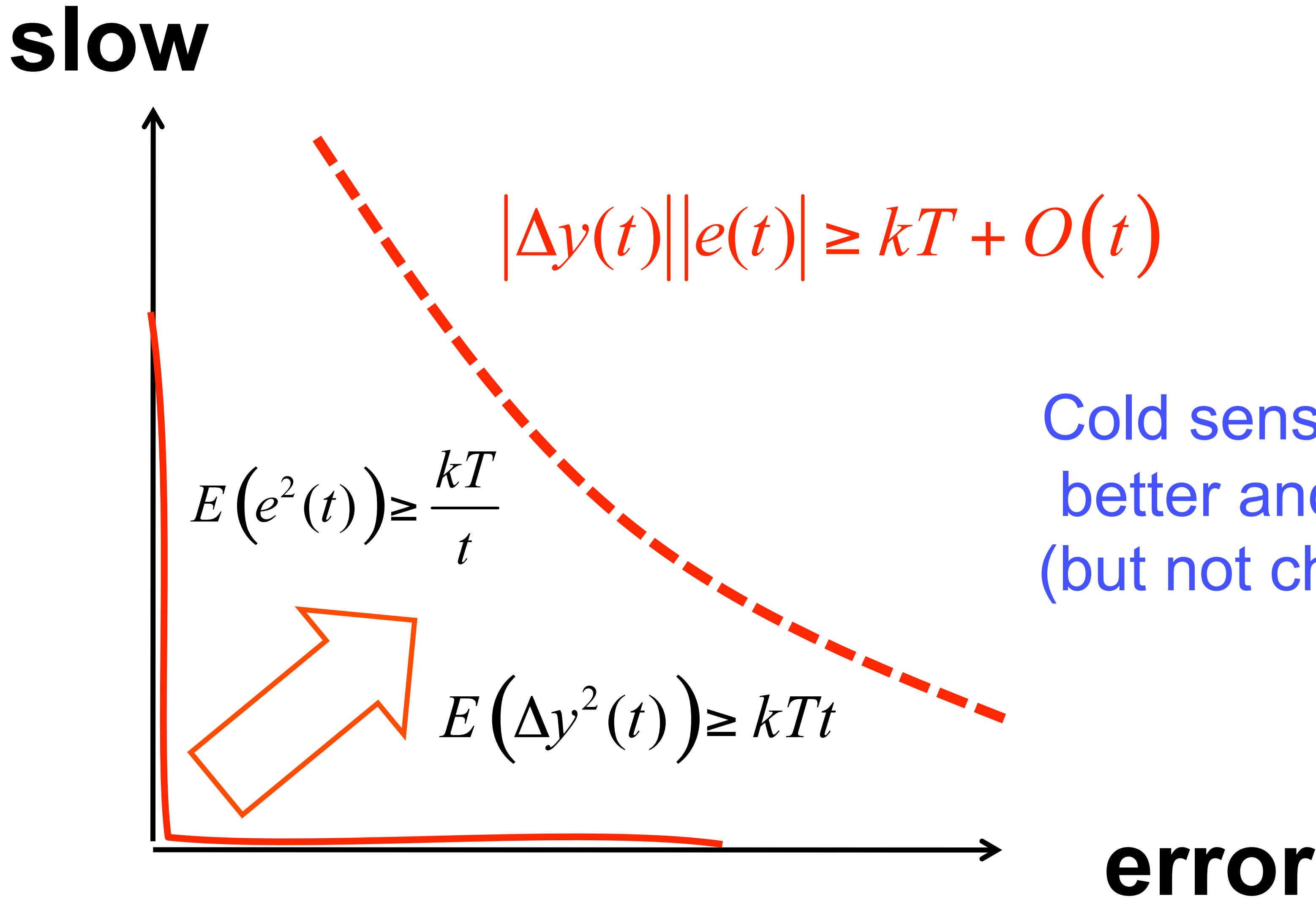
$$E\left(e^2(t)\right) \geq \frac{kT}{t} + O(1)$$

Sensor “noise”

- Simplest hard tradeoffs on speed and errors
- More tradeoffs (e.g. energy overhead vs speed vs errors)
- Just scratching the surface
- Actuators, computation, quantum effects,…?

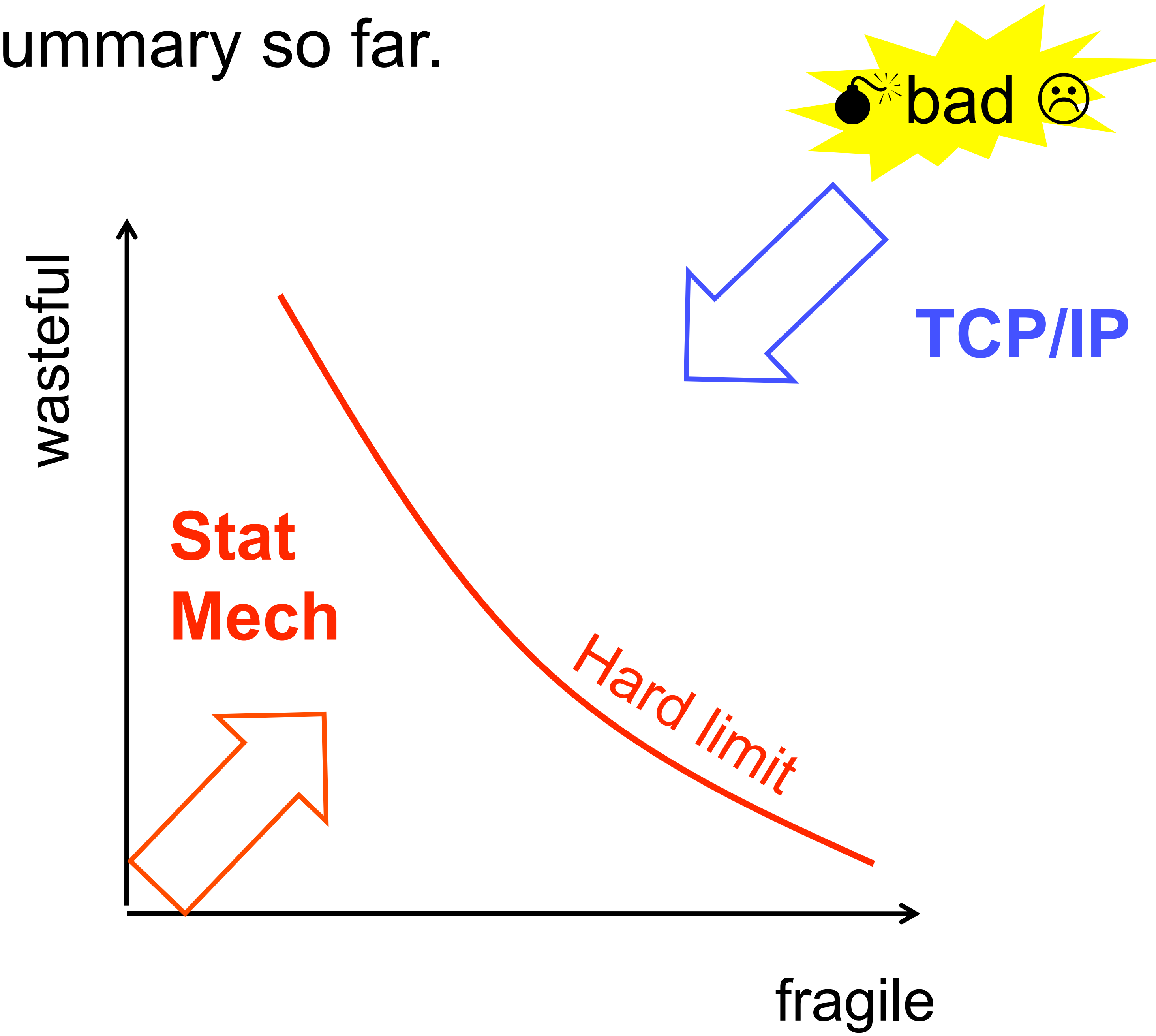
- Aside: linear active elements need nonlinear implementation

A transient and far-from-equilibrium upgrade of statistical mechanics

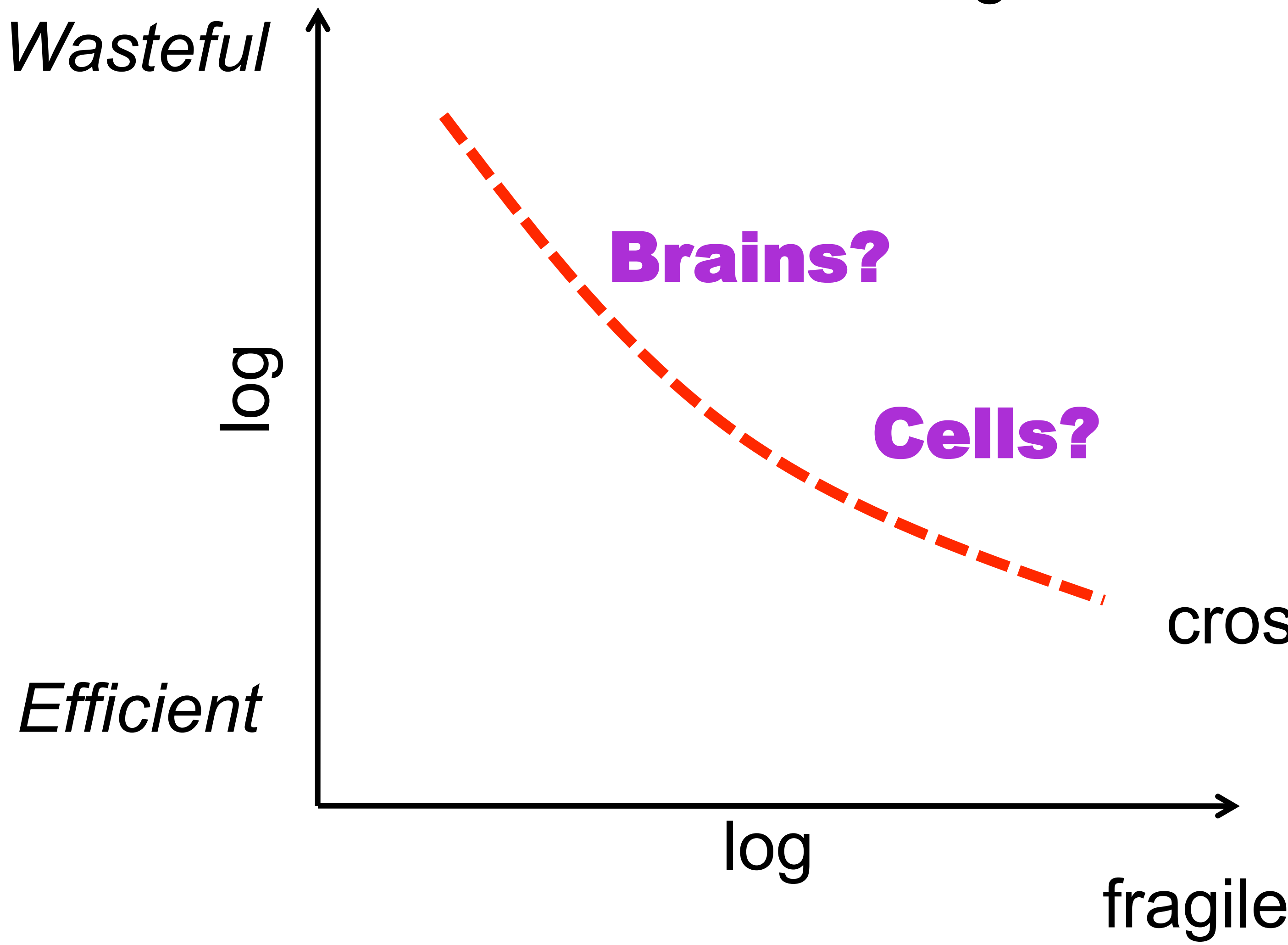


Cold sensors are better and faster (but not cheaper)

Summary so far.



Conjecture: Cells and brains are RYF but not gratuitously fragile



They avoid cross-layering?