Complex Network Architecture: A cartoon guide

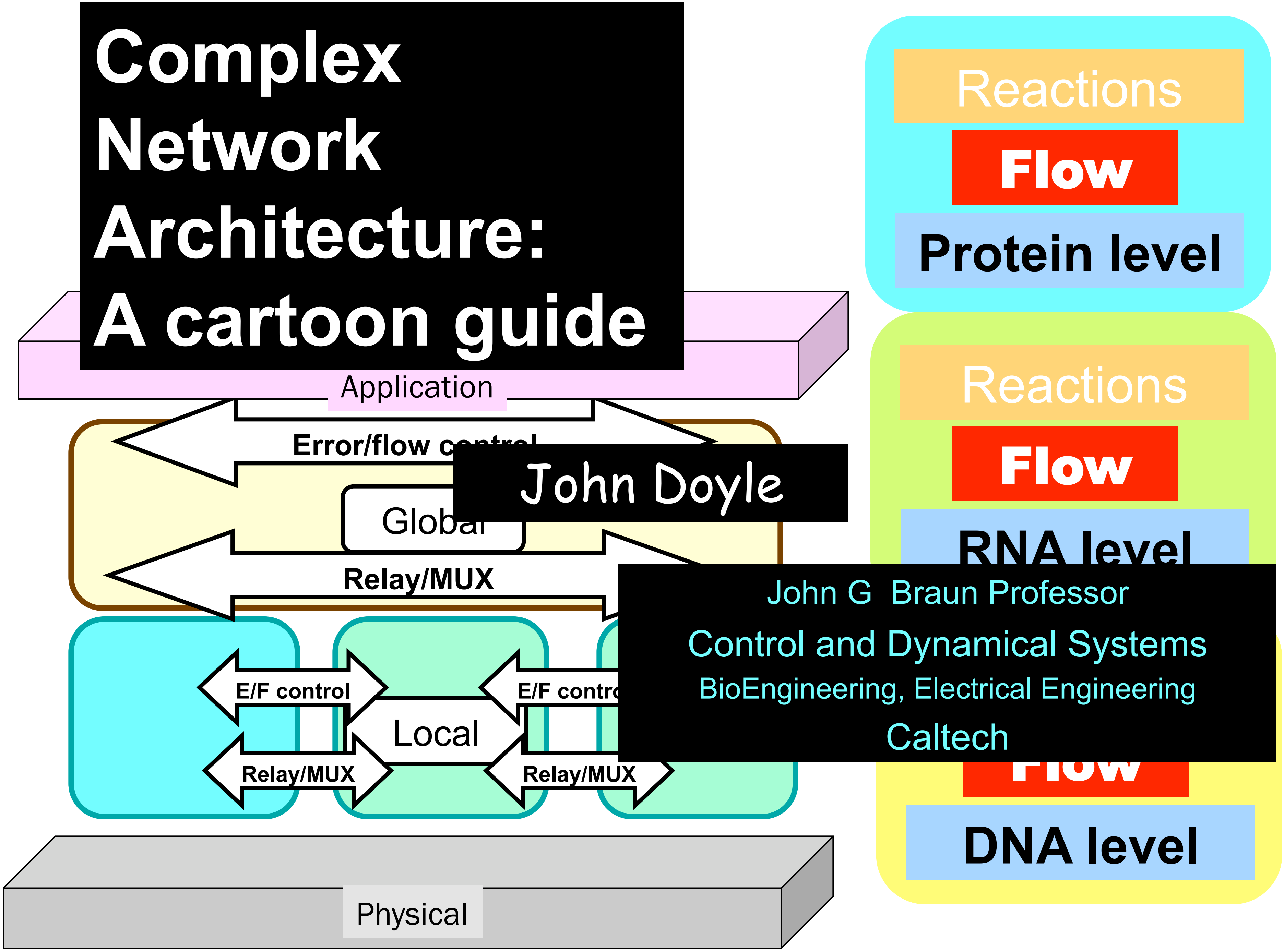
John Doyle

John G Braun Professor

Control and Dynamical Systems

BioEngineering, Electrical Engineering

Caltech



Outline

- Not merely “complexity, networks, abstraction, recursion, modularity,…”
- But very specific forms of these that are needed for networks
- Emphasize fundamentals
- Illustrate with case studies and cartoons:
  - Internet versus bacterial biosphere
  - Operating systems
  - Global brain architecture
  - Smartgrid and cyberphysical

Next steps?

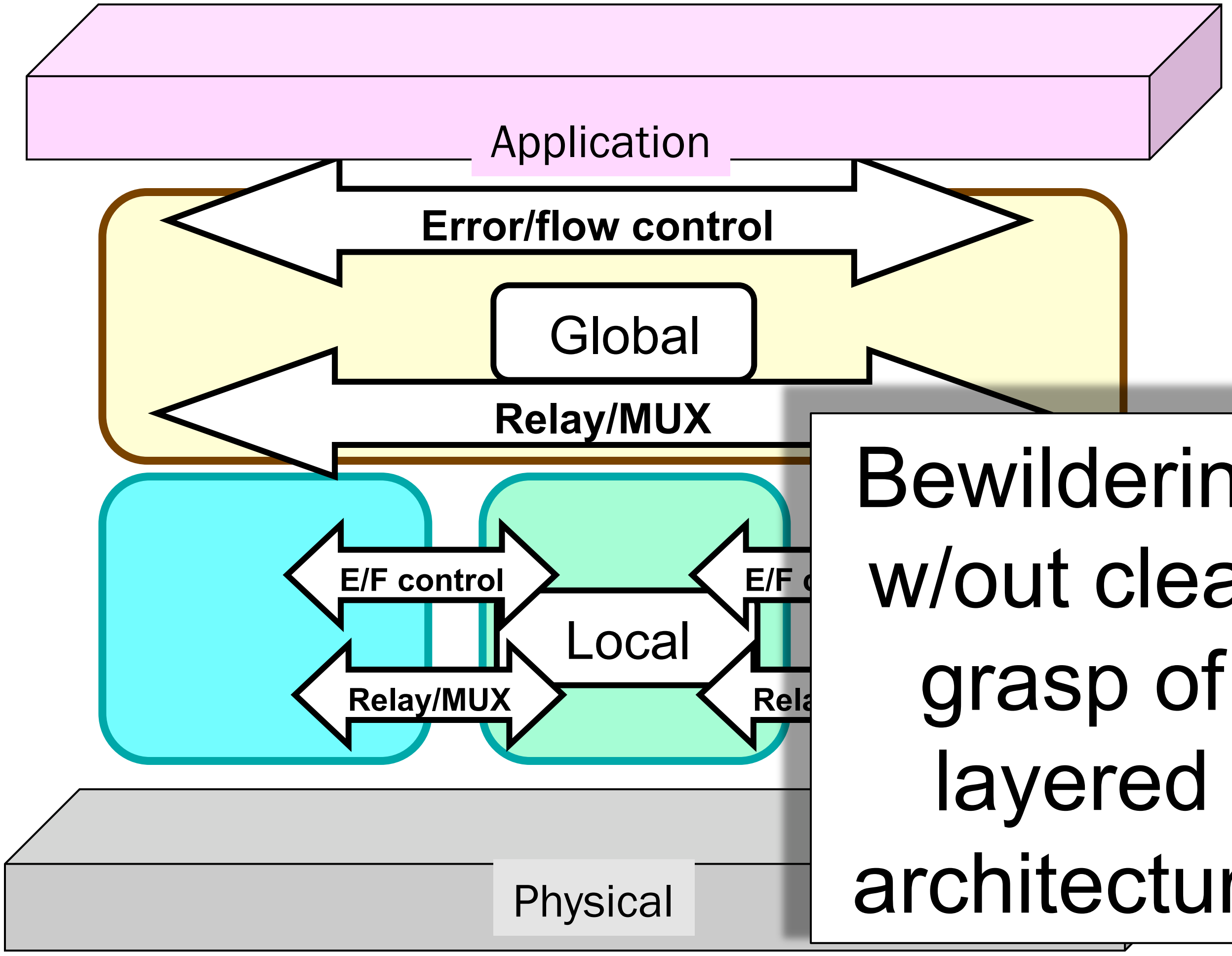
- New course this term? (CDS 213?)
- Discuss at 1pm
- Flesh out details
- Integrated control, comms, computing, thermo/statmech, optimization, games, etc
- Motivated by very generic network challenges

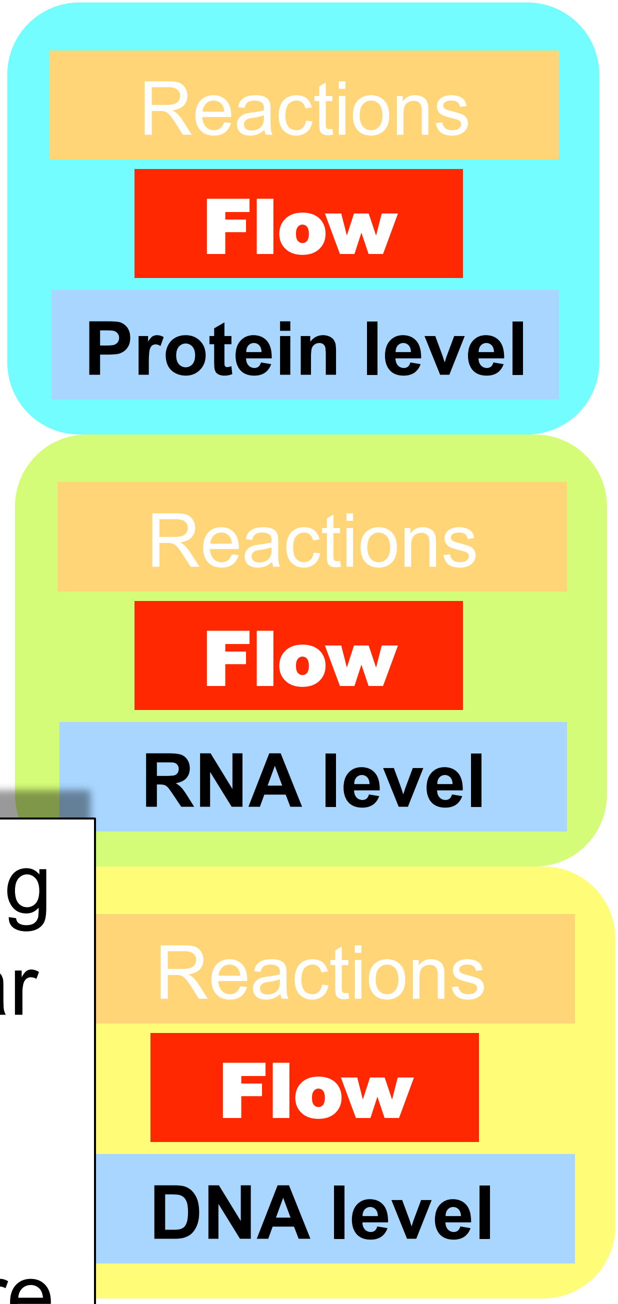
Network Math and Engineering (NetME) Challenges

- Predictive modeling, simulation, and analysis of complex systems in technology and nature
- Theoretical foundation for design of network architectures
- Balance rigor/relevance, integrative/coherent
- Model/simulate is critical but limited
  - Predicting rare but catastrophic events
  - Design, not merely analysis
  - Managing complexity and uncertainty

Huge range of dynamics

- Spatial
- Temporal





Bewildering w/out clear grasp of layered architecture

“Architecture”

- Most persistent, ubiquitous, and global features of organization
- Constrains what is possible for good or bad
- Platform that enables (or prevents) innovation, sustainability, etc,
- Internet, biology, energy, manufacturing, transportation, water, food, waste, law, etc
- Existing architectures are unsustainable
- Theoretical foundation is fragmented, incoherent, incomplete

Lots of Influences Not Referenced Properly in These Notes

- Biology/Medicine (Savageau, G&K, Mattick, Csete, Arkin, Alon, Caporale, de Duve, Exerc Physio, Acute Care, etc etc…)
- Internet (Kelly/Low, Willinger, Clark, Wroclawski, Day, Chang, etc etc)
- Architecture (Alexander, Salingeros,…)
- Aerospace (many, Maier is a good book)
- Philosophy/History (Fox Keller, Jablonka&Lamb)
- Physics/ecology (Carlson)
- Management (Baldwin,…)
- Resilience/Safety/Security Engineering/Economics (Wood, Anderson, Leveson, …)

Infrastructure networks?

- Power
- Transportation
- Water
- Waste
- Food
- Healthcare
- Finance

All examples of “bad” architectures:
- Unsustainable
- Hard to fix

Where do we look for “good” examples?

Informative case studies in architecture

- Internet and related technology (OS)
- Systems biology (particularly, bacterial biosphere)
- System medicine and physiology
- Ecosystems (e.g. So Cal wildfire ecology)
- Aerospace systems
- Electronic Design Autom. (Platform Based Design)

- Multiscale physics (turbulence, stat mech)
- Misc: buildings/cities, Lego, clothing/fashion, barter/markets/money/finance, social/political

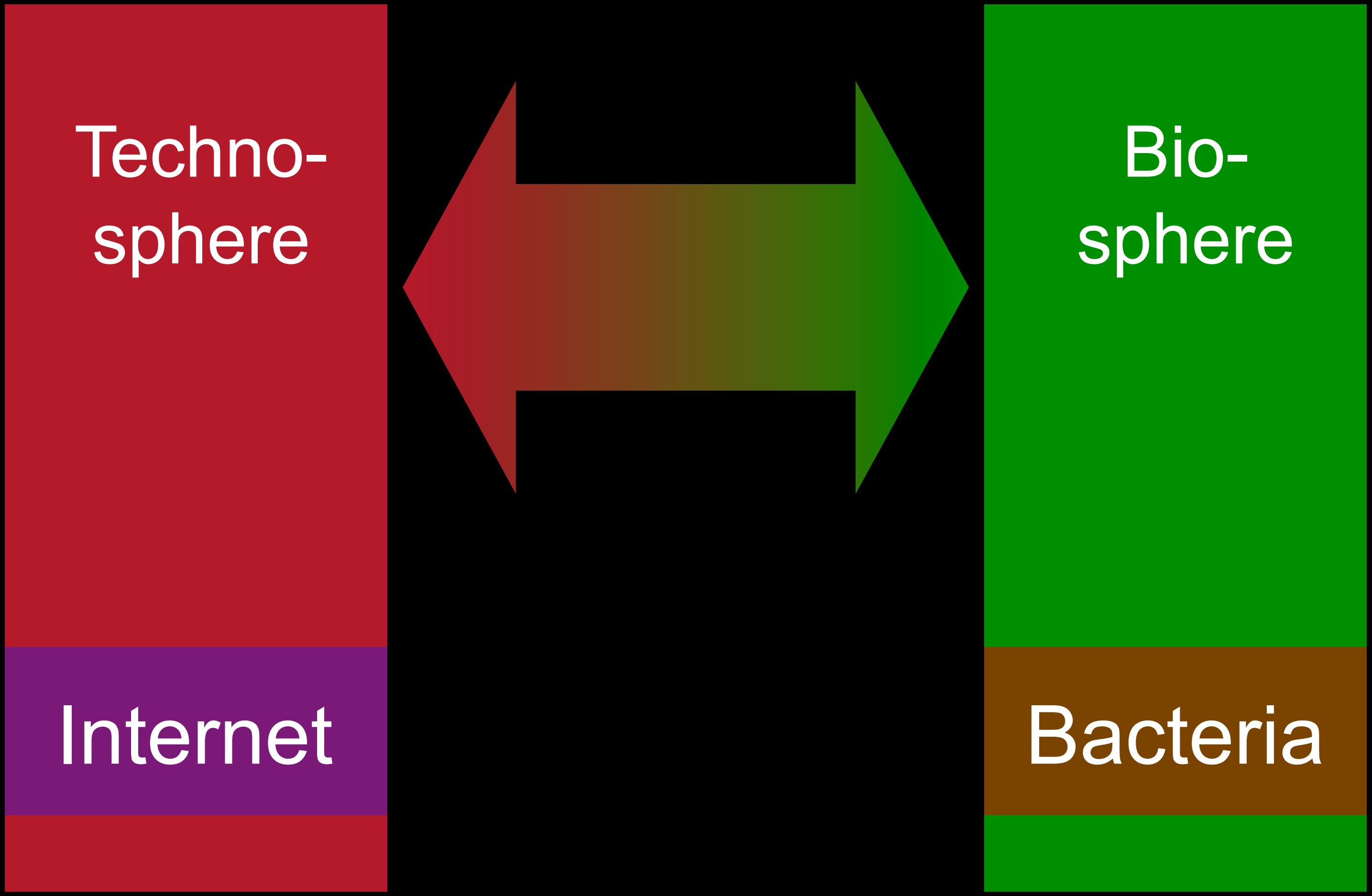
- Successful architectures
- Robust, evolvable
- Universal, foundational
- Accessible, familiar
- Unresolved challenges
- New theoretical frameworks
- Boringly retro?

Simplest case studies

Internet

Bacteria

- Universal, foundational



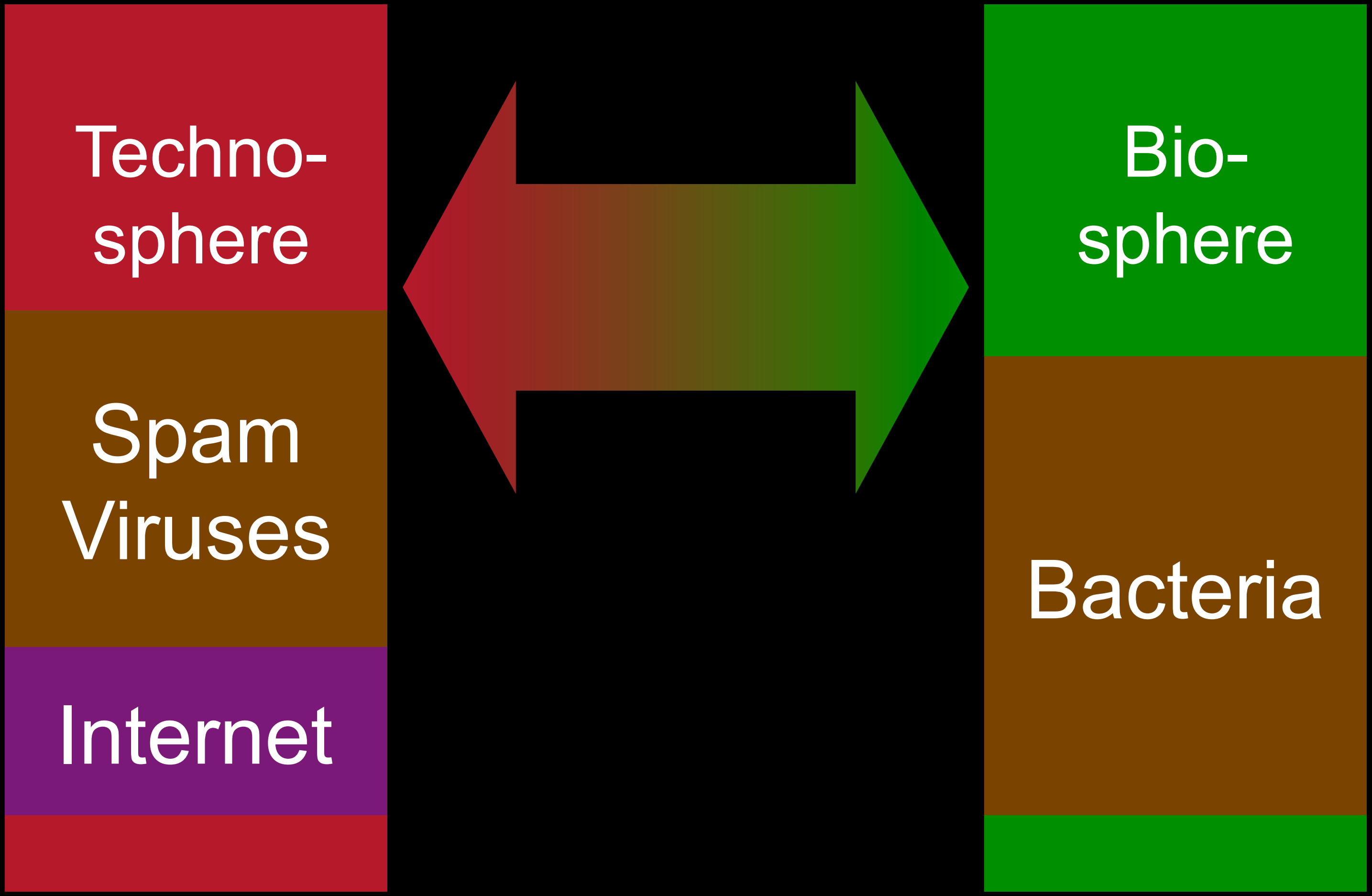
Technosphere

Internet

Biosphere

Bacteria

- Universal, foundational



Techno-sphere

Spam
Viruses

Internet

Bio-sphere

Bacteria

Two lines of research:

1. Patch the existing Internet architecture so it handles its new roles

Techno-sphere

Internet

- Real time
- Control ***over*** (not just of) networks
- Action in the physical world
- Human collaborators and adversaries
- Net-centric ***everything***

Modern theory and the Internet

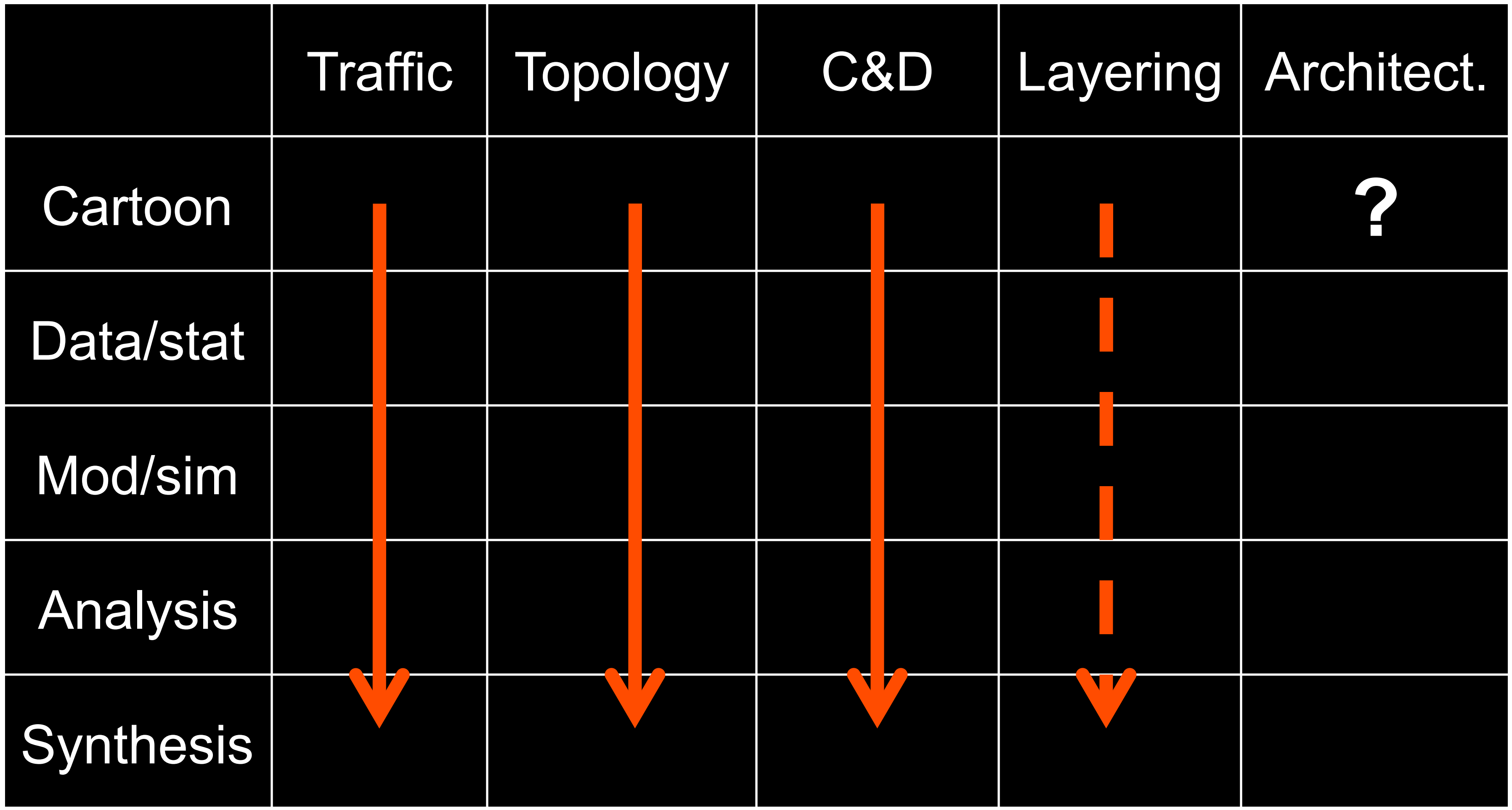
Levels of understanding

- Verbal/cartoon
- Data and statistics
- Modeling and simulation
- Analysis
- Synthesis

Topics

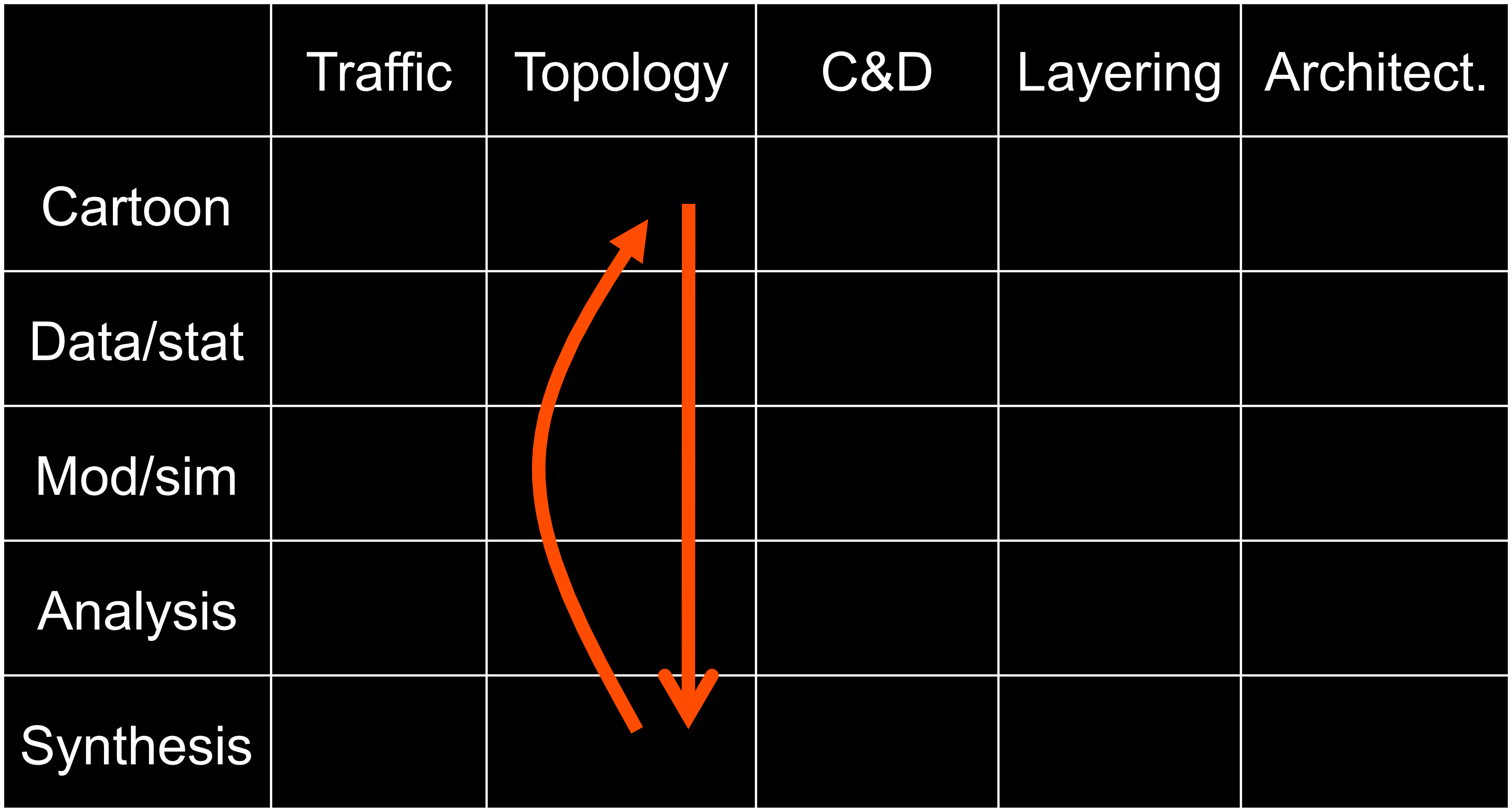
- Traffic
- Topology
- Control and dynamics
- Layering
- Architecture

Recent progress (1995-)



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

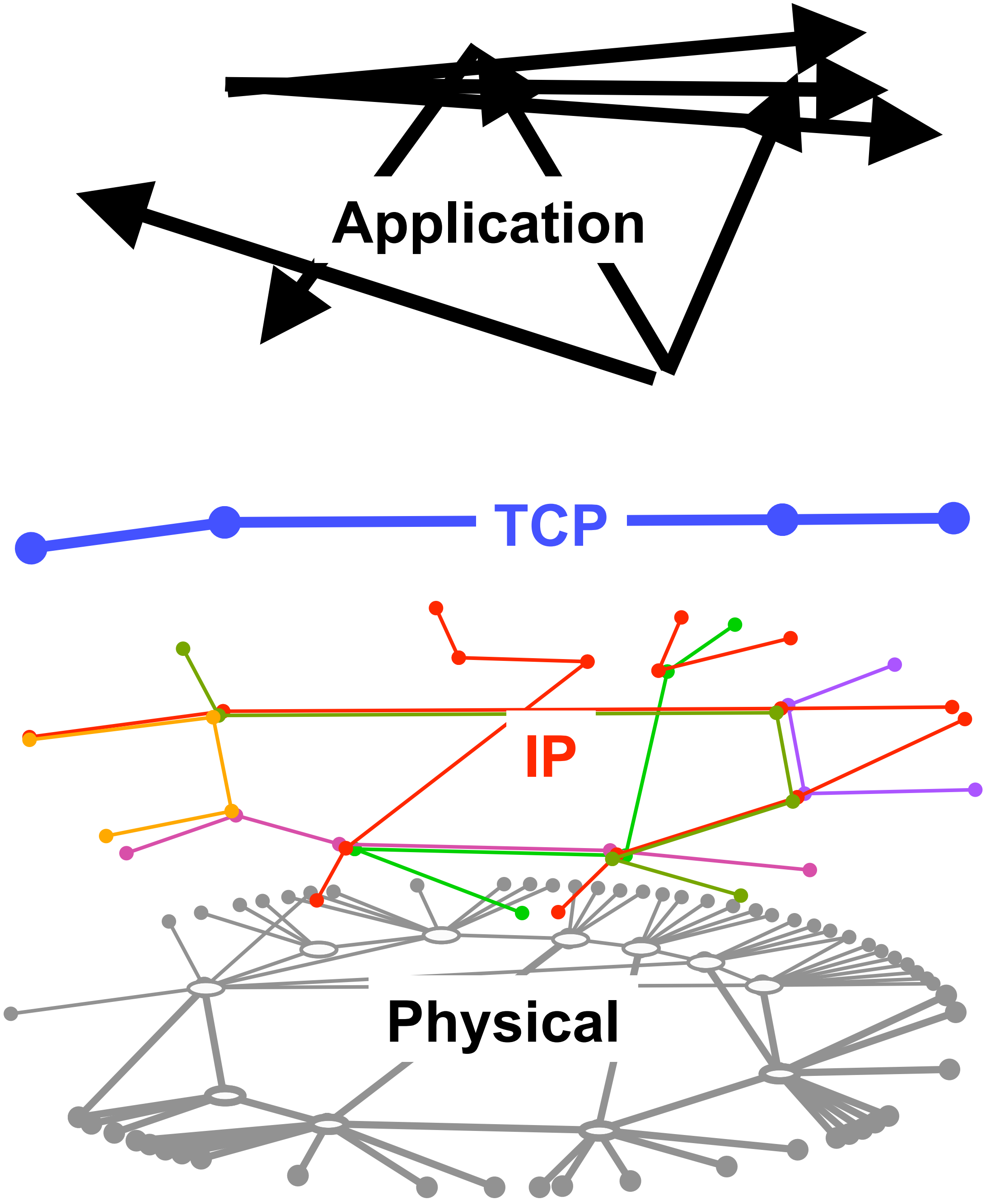
Recent progress

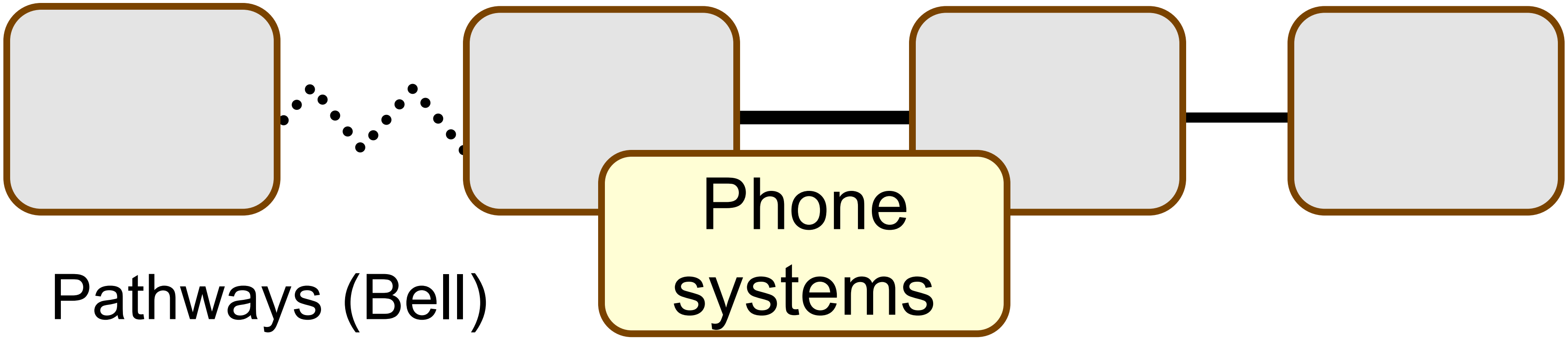


	Traffic	Topology	C&D	Layering	Architect.
Cartoon					
Data/stat					
Mod/sim					
Analysis					
Synthesis					

Architecture is ***not*** graph topology.

Architecture facilitates arbitrary graphs.



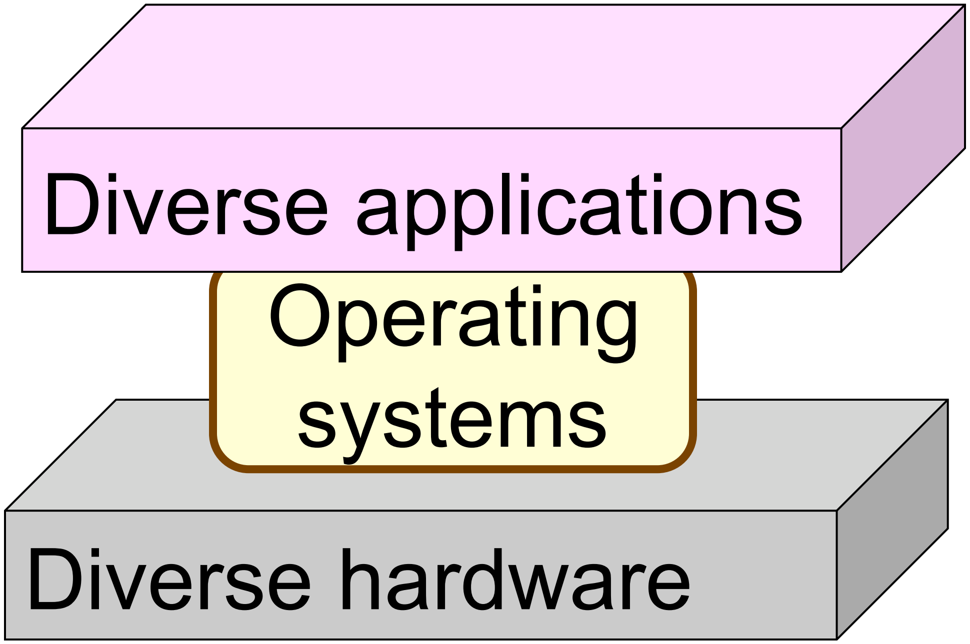


Phone
systems

Pathways (Bell)
Communications

Layers (Net)
Computer

Ancient network
architecture:
“Bell-heads
versus
Net-heads”

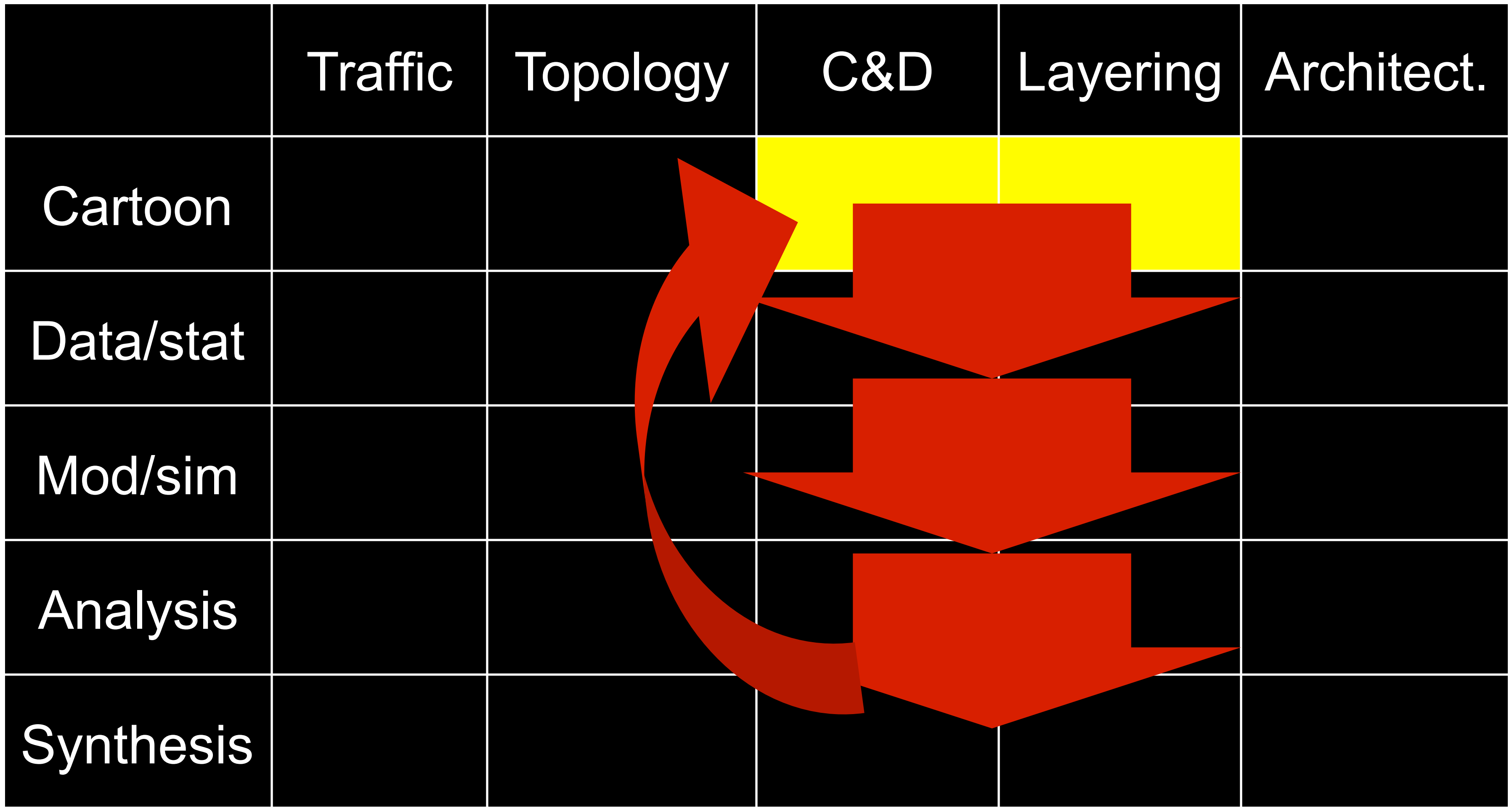


Diverse applications

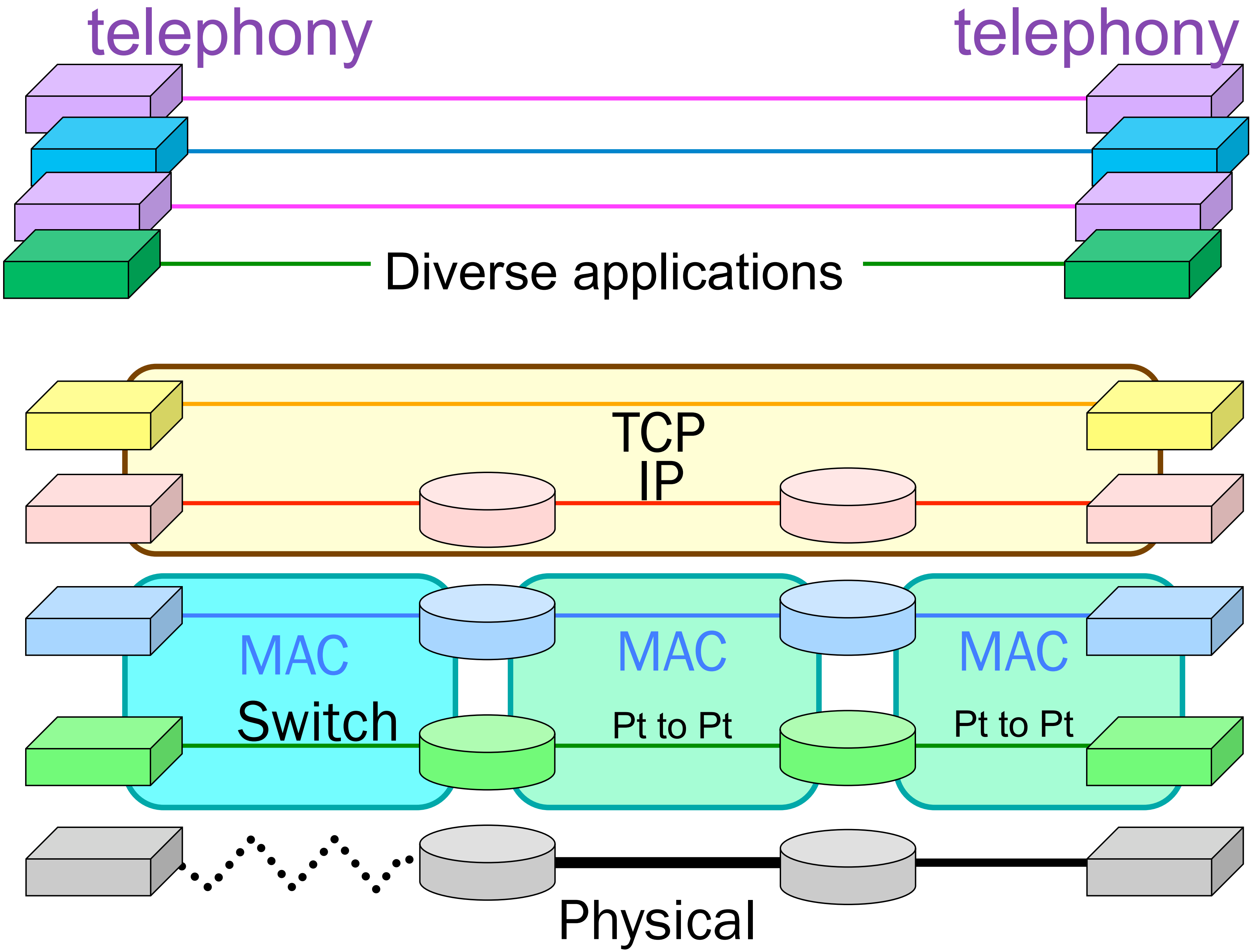
Operating
systems

Diverse hardware

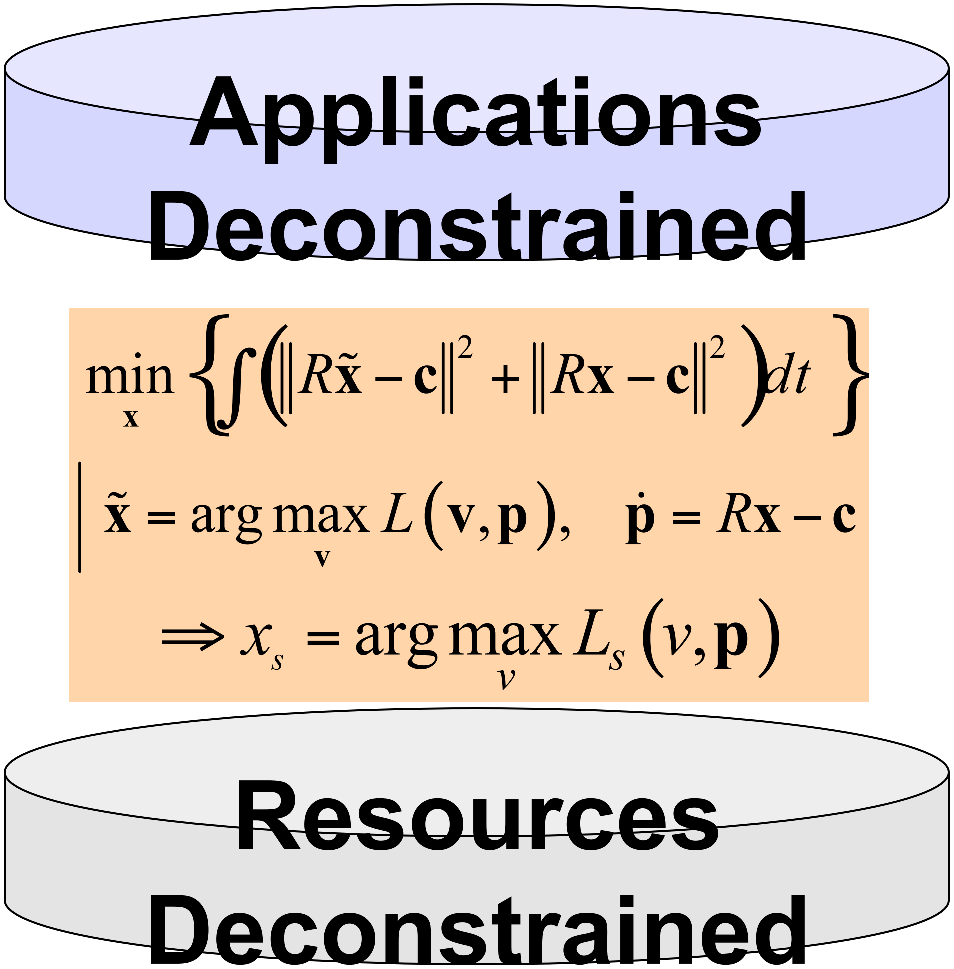
Recent progress (1995-)



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


telephony
telephony
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

- Layering as optimization decomposition
- Optimal control
- Robust control
- Game theory
- Network coding
- Recursive layers?

Cyber-Physical Theories

- Thermodynamics
- Communications
- Control
- Computation

Cyber

- Thermodynamics
- Communications
- Control
- Computation

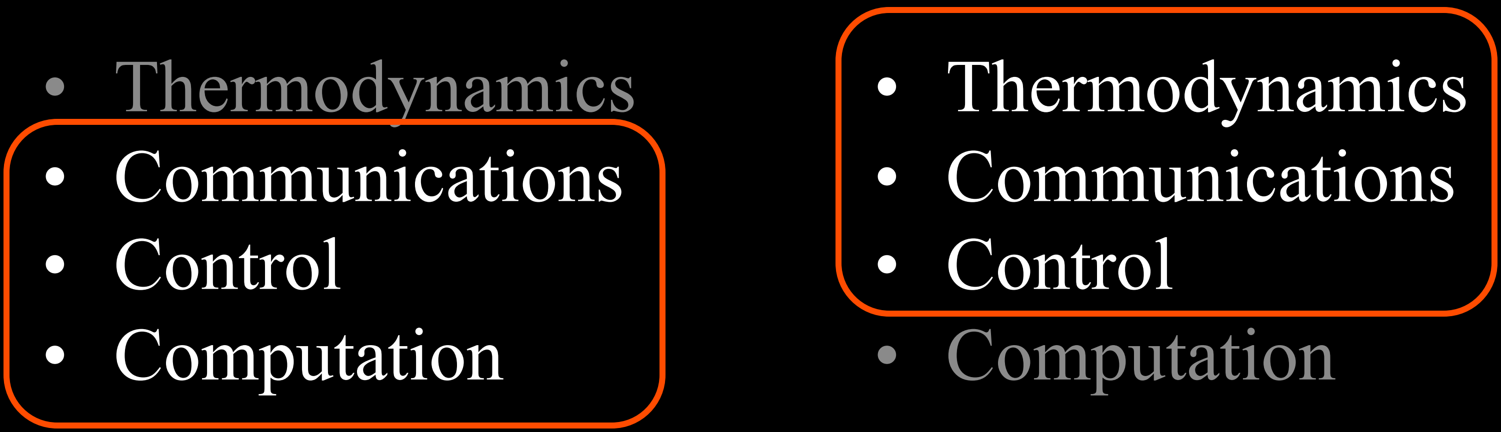
Internet

Physical

- Thermodynamics
- Communications
- Control
- Computation

Bacteria

Case studies



Cyber

- Thermodynamics
- Communications
- Control
- Computation

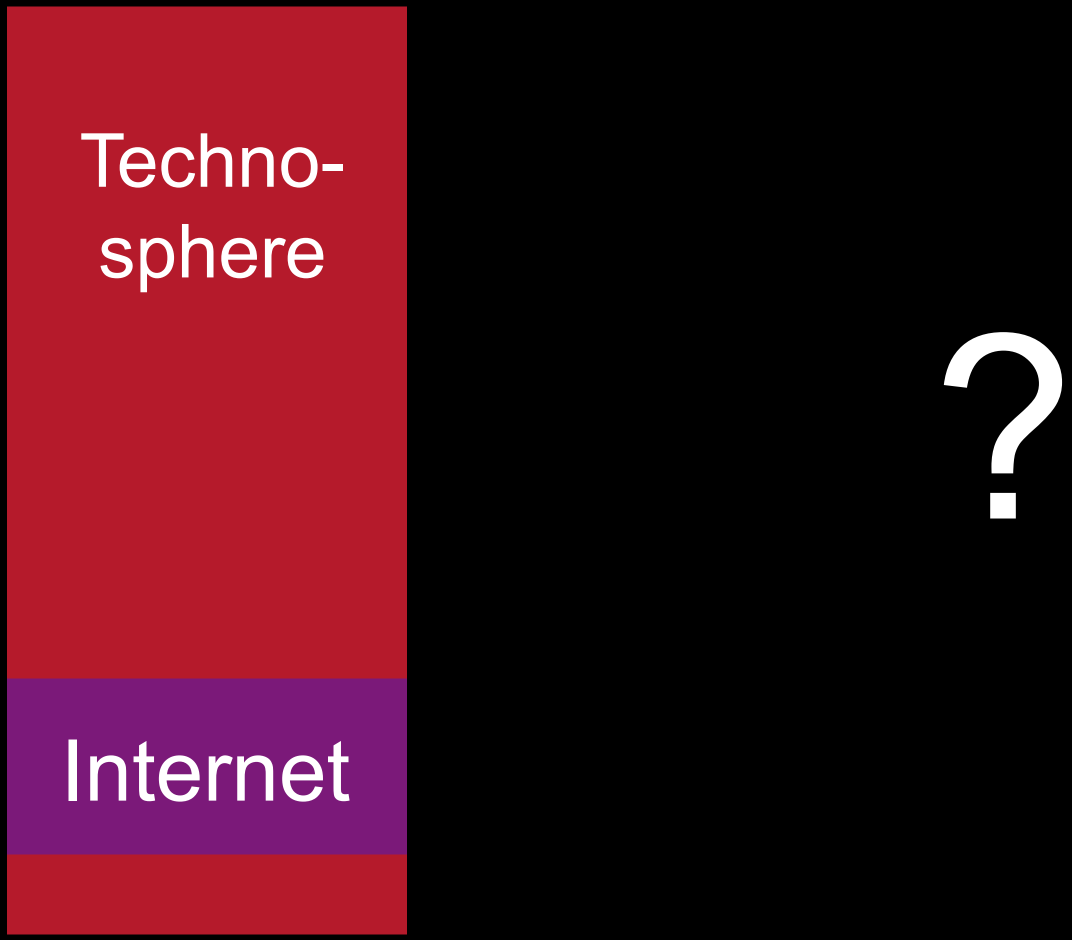
Physical

- Thermodynamics
- Communications
- Control
- Computation

Promising unifications

Two lines of research:

1. Patch the existing Internet architecture
2. Fundamentally rethink network architecture

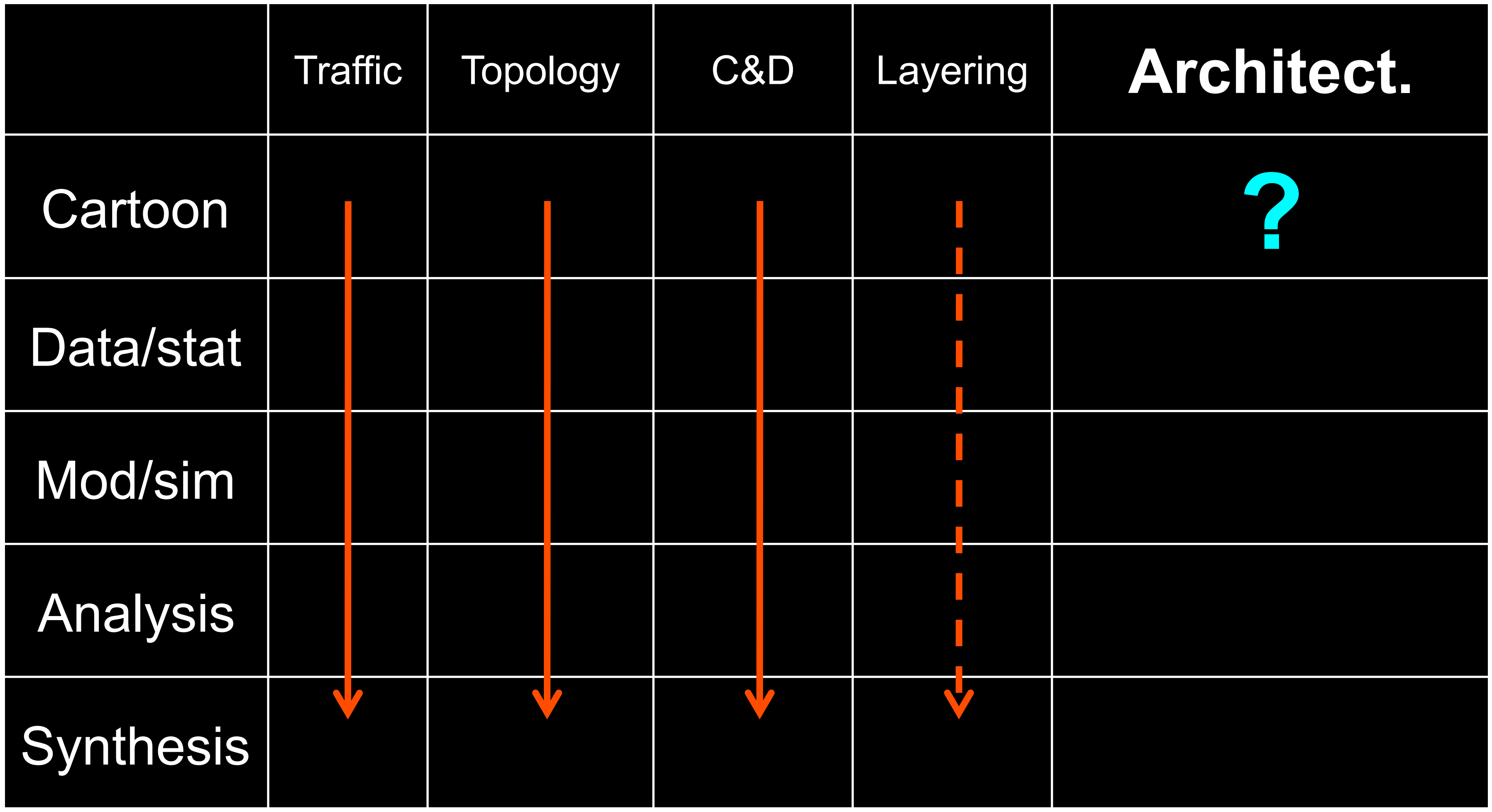


Techno-sphere

Internet

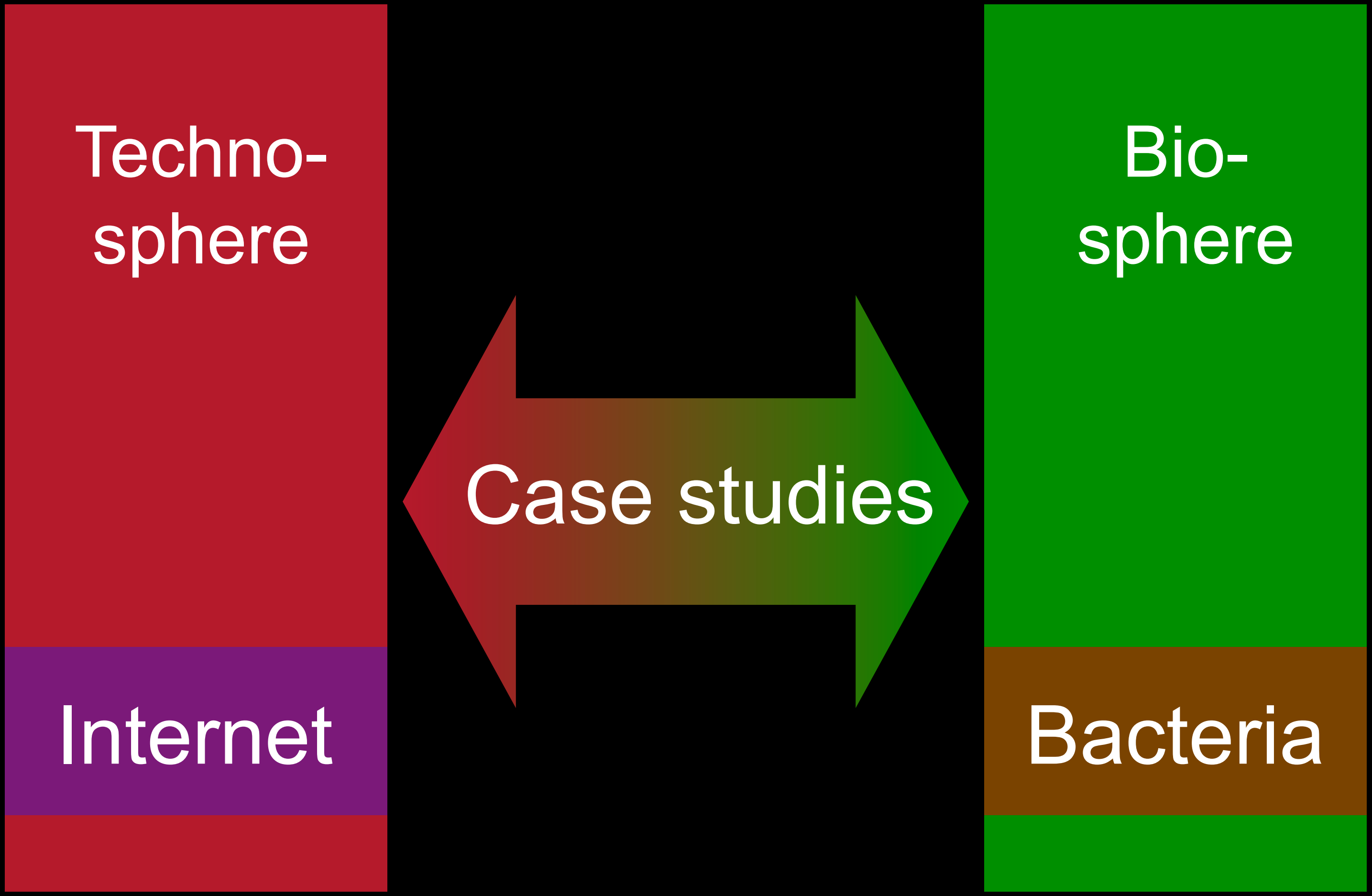
?

Architecture?



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

1.
2. Fundamentally rethink network architecture



Techno-sphere

Internet

Case studies

Bio-sphere

Bacteria

Biology versus the Internet

Similarities

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

Differences

- Metabolism
- Materials and energy
- Autocatalytic feedback
- Feedback complexity
- Development and regeneration
- >4B years of evolution
- How the parts work?

Biology versus the Internet

Similarities

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

Differences

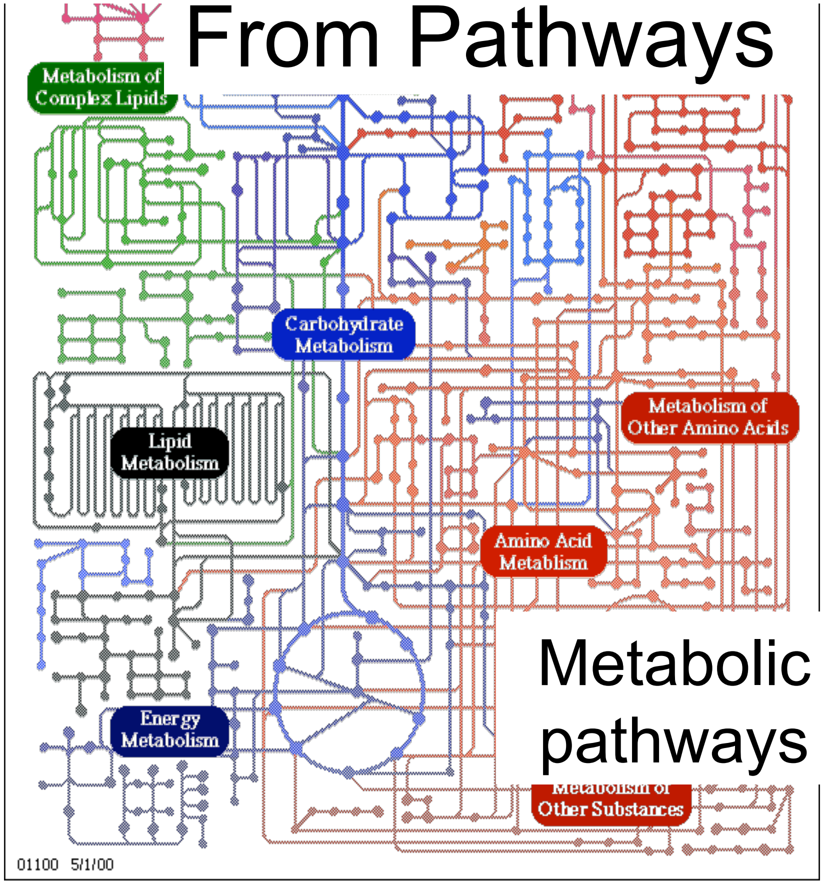
- **Metabolism**
- **Materials and energy**
- Autocatalytic feedback
- Feedback complexity
- Development and regeneration
- >4B years of evolution

Focus on
bacterial biosphere

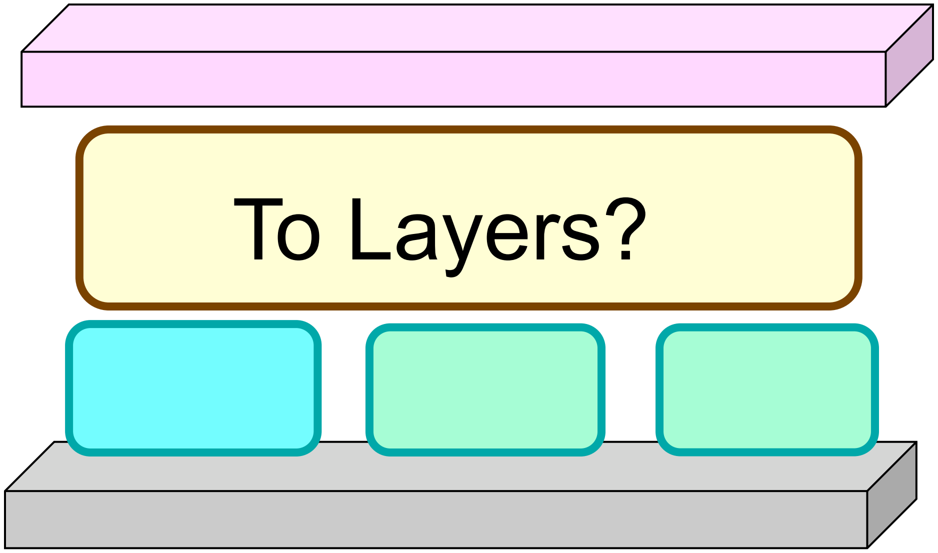
“Central dogma”

Network architecture?

From Pathways

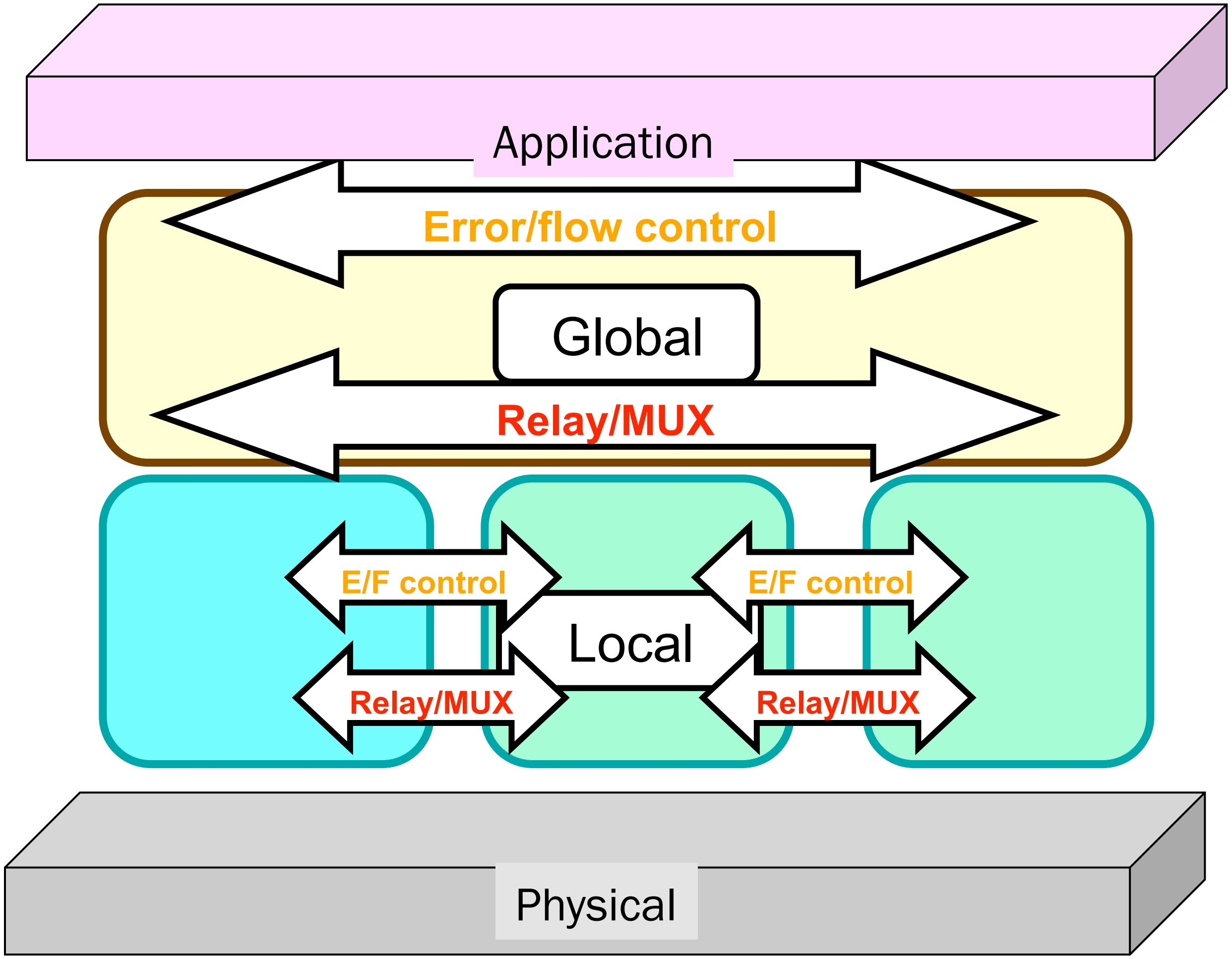


Metabolic pathways



To Layers?

Recursive control structure



Application

Error/flow control

Global

Relay/MUX

E/F control

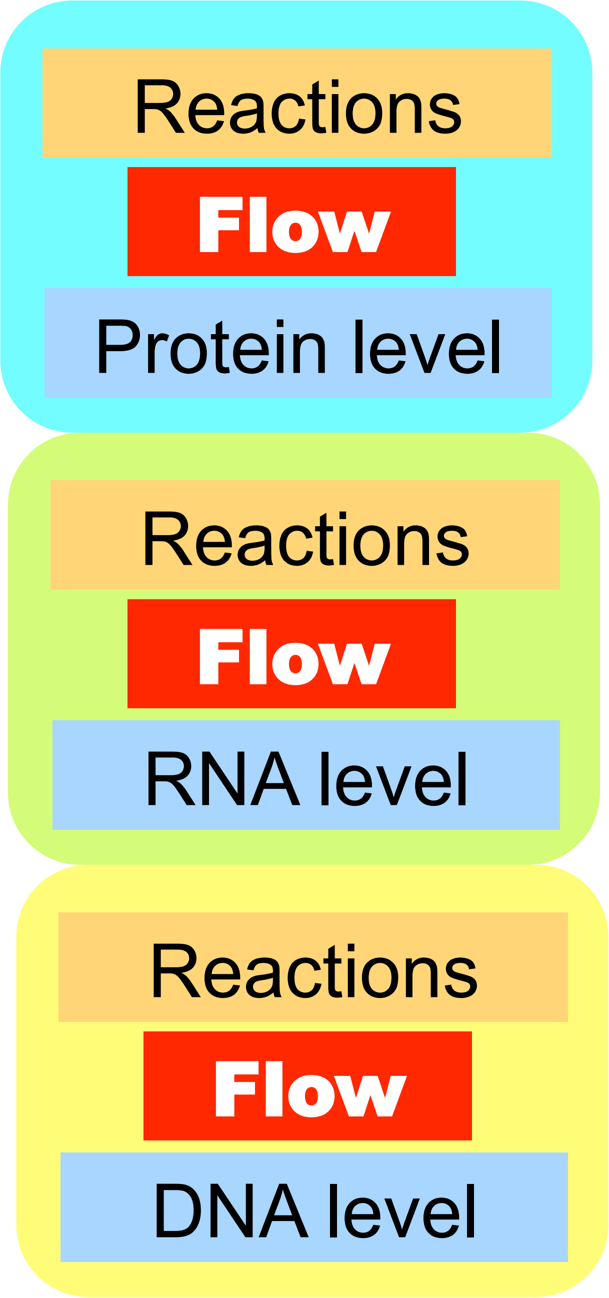
E/F control

Local

Relay/MUX

Relay/MUX

Physical



Reactions

Flow

Protein level

Reactions

Flow

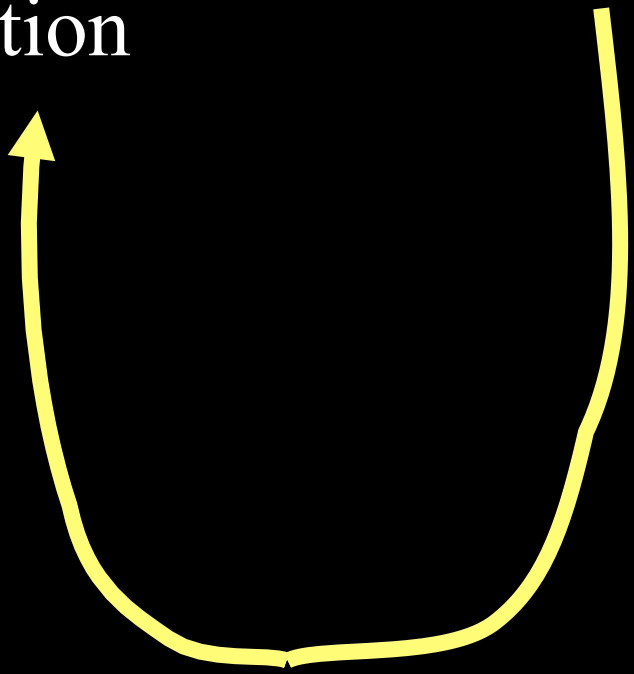
RNA level

Reactions

Flow

DNA level

In the real (vs virtual) world



What matters:

- Action

What doesn’t:

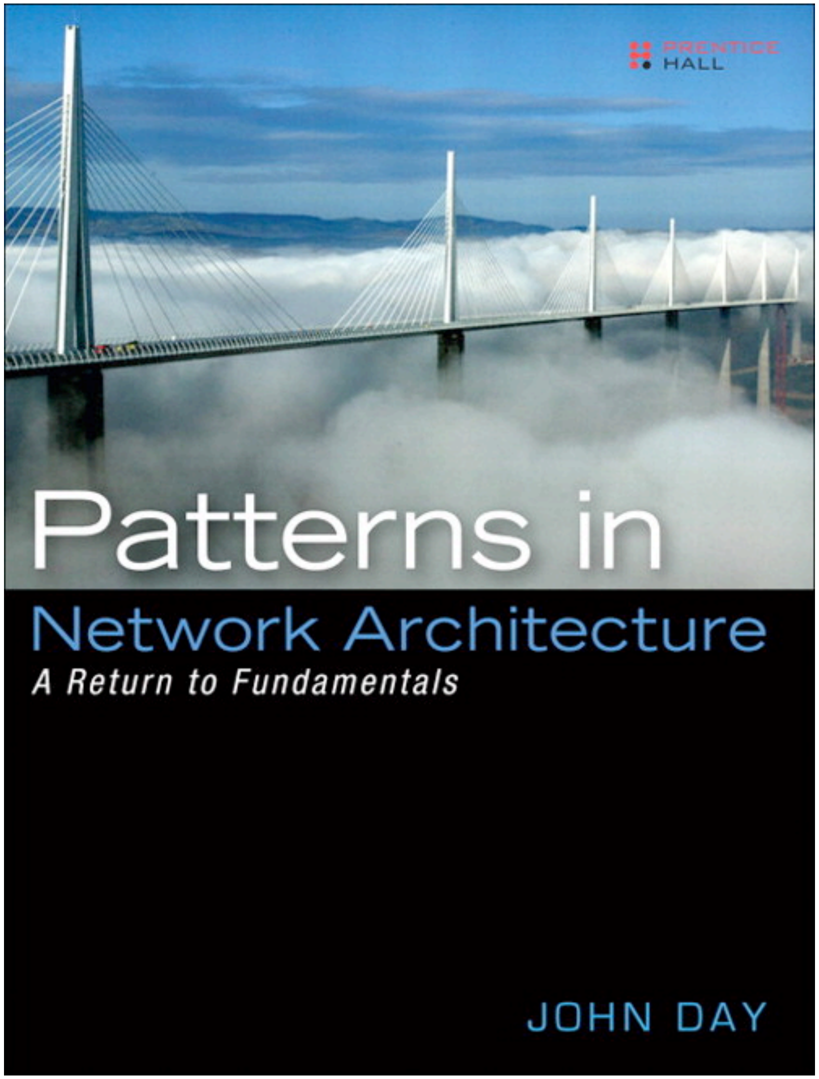
- Data
- Information
- Computation
- Learning
- Decision
- …

Two lines of research:

1. Patch the existing Internet architecture
2. Fundamentally rethink network architecture

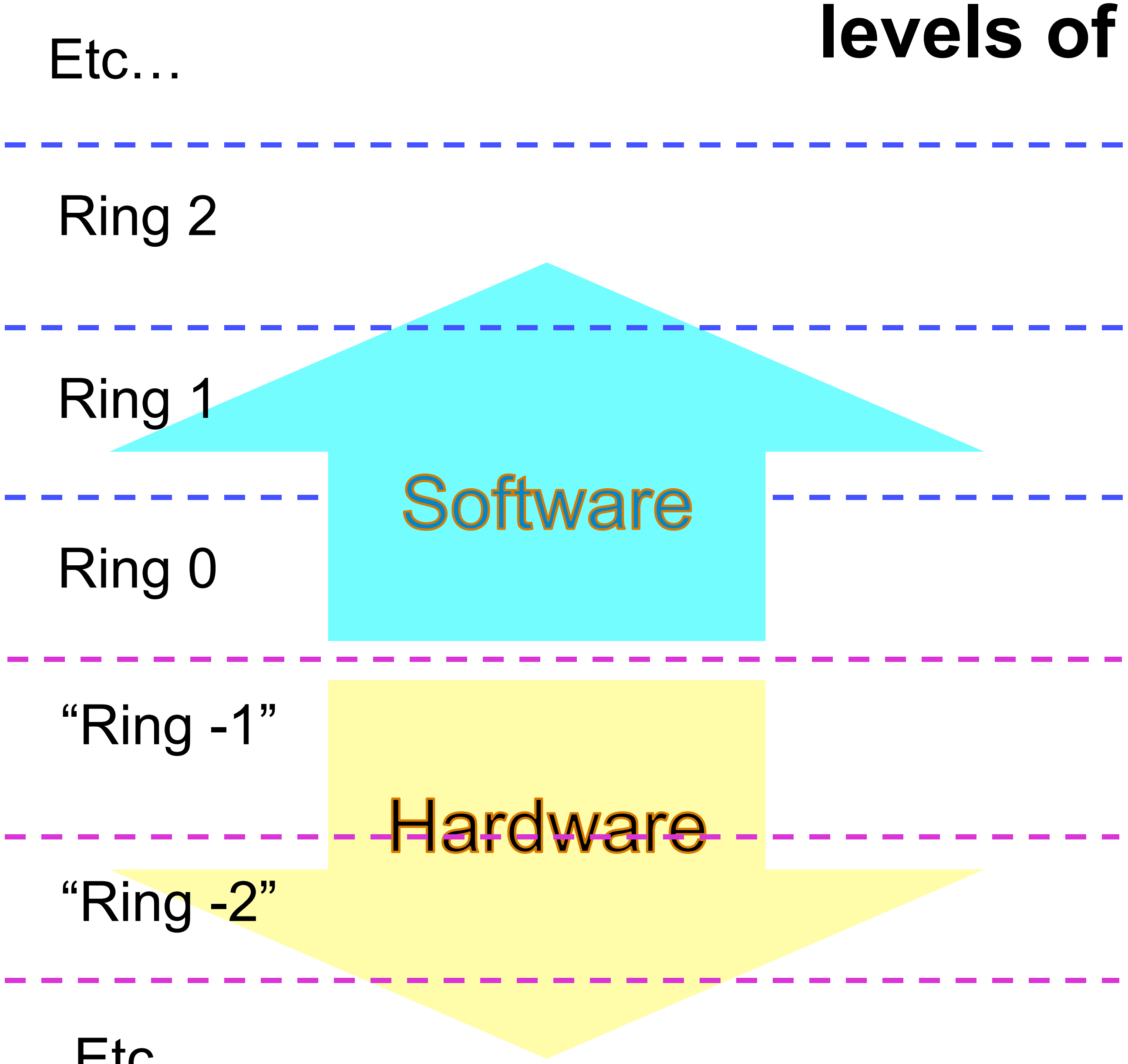
?

PNA



“return to fundamentals”

“Rings” are HW defined levels of “protection”



Etc…

Ring 2

Ring 1

Ring 0

“Ring -1”

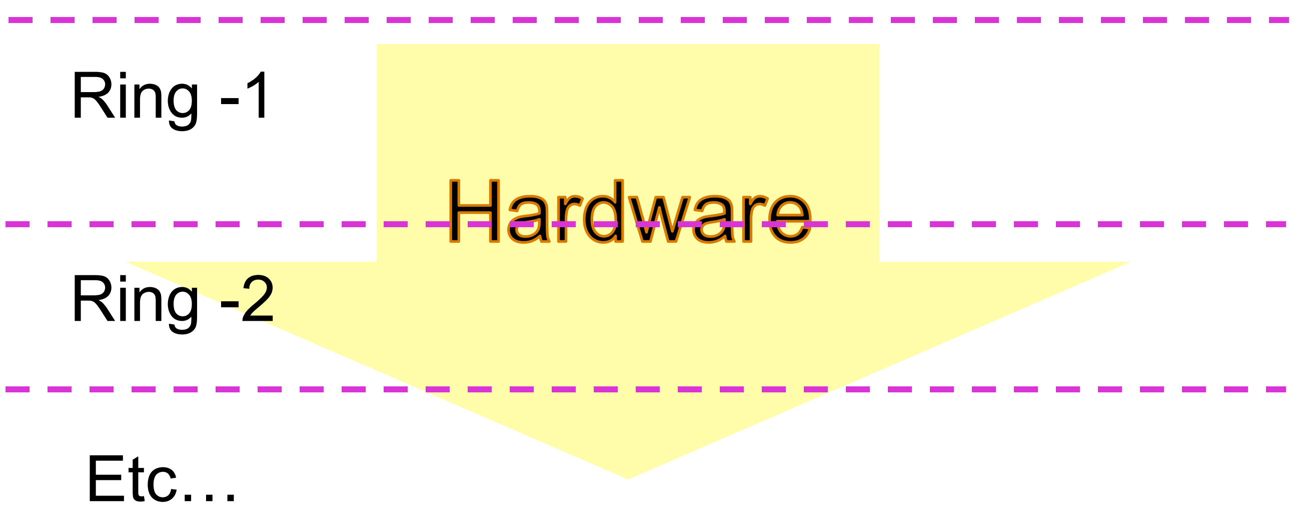
“Ring -2”

Etc…

Platform Based Design

- Negative rings don't mean the same thing
- They would correspond to abstraction layers in hardware design
- We will temporarily defer HW, but..
- PBD (Platform Based Design) is very compatible
- Often a key design issue in PBD is where to put the HW/SW boundary
- The PNA view of layering can be viewed as a special case of PBD

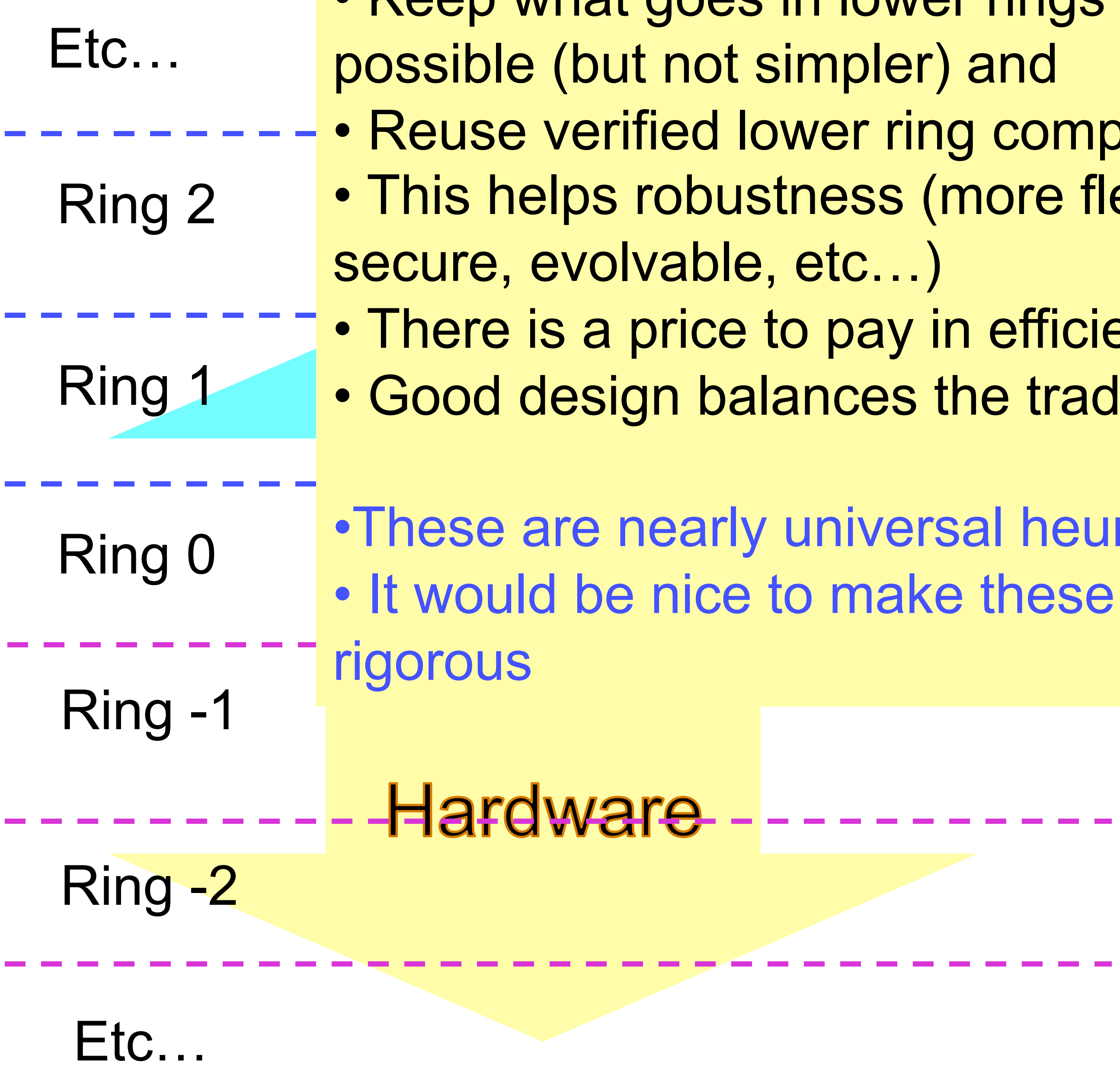
Ring 0



Design heuristics (KISS or E2E)

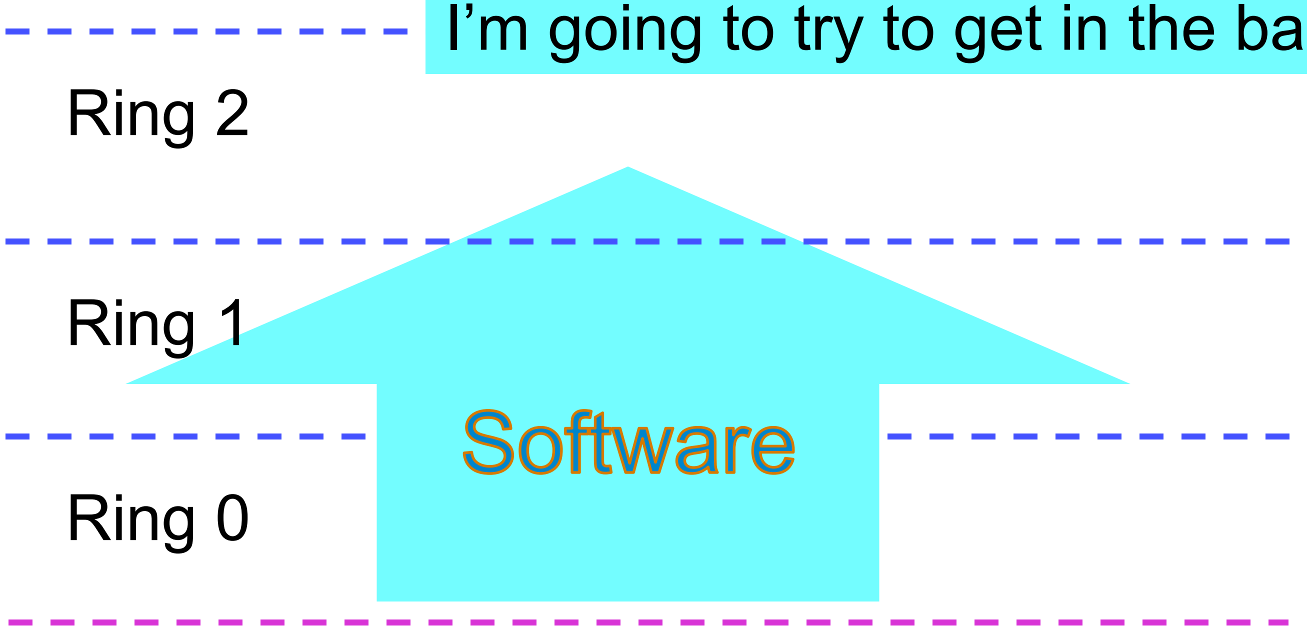
- Keep what goes in lower rings as simple as possible (but not simpler) and
- Reuse verified lower ring components…
- This helps robustness (more flexible, verifiable, secure, evolvable, etc…)
- There is a price to pay in efficiency
- Good design balances the tradeoff

- These are nearly universal heuristics
- It would be nice to make these heuristics more rigorous

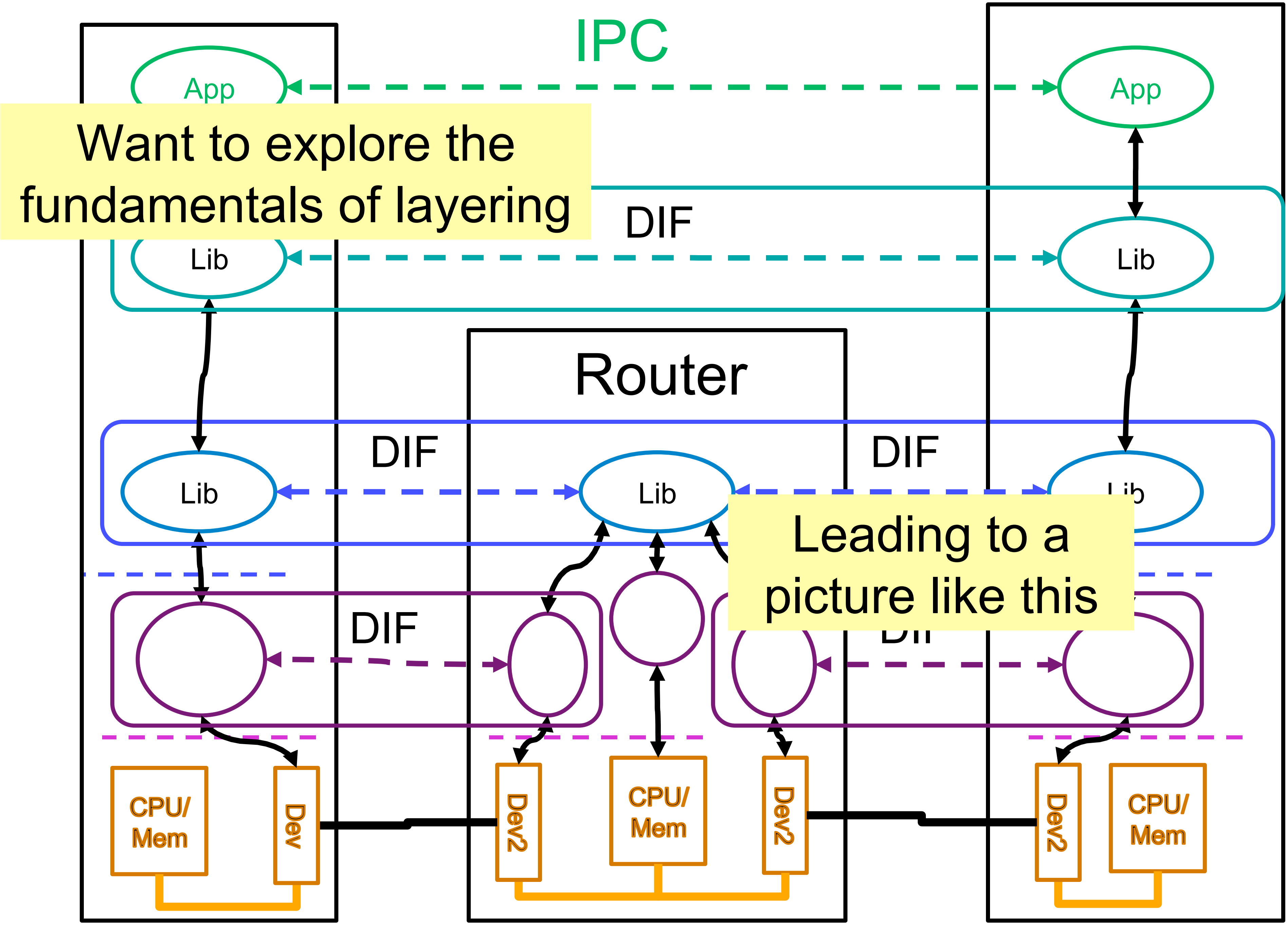


My first mistake…

I’m only going to draw 3 rings of software and I’m not going to put things in the right rings, but I’m going to try to get in the ballpark…



- I’m not going to do a very good job of drawing the HW
- Actually I won’t do a good job of drawing anything but I think the hardware will be really bad.
- No rings of hardware.



Want to explore the fundamentals of layering

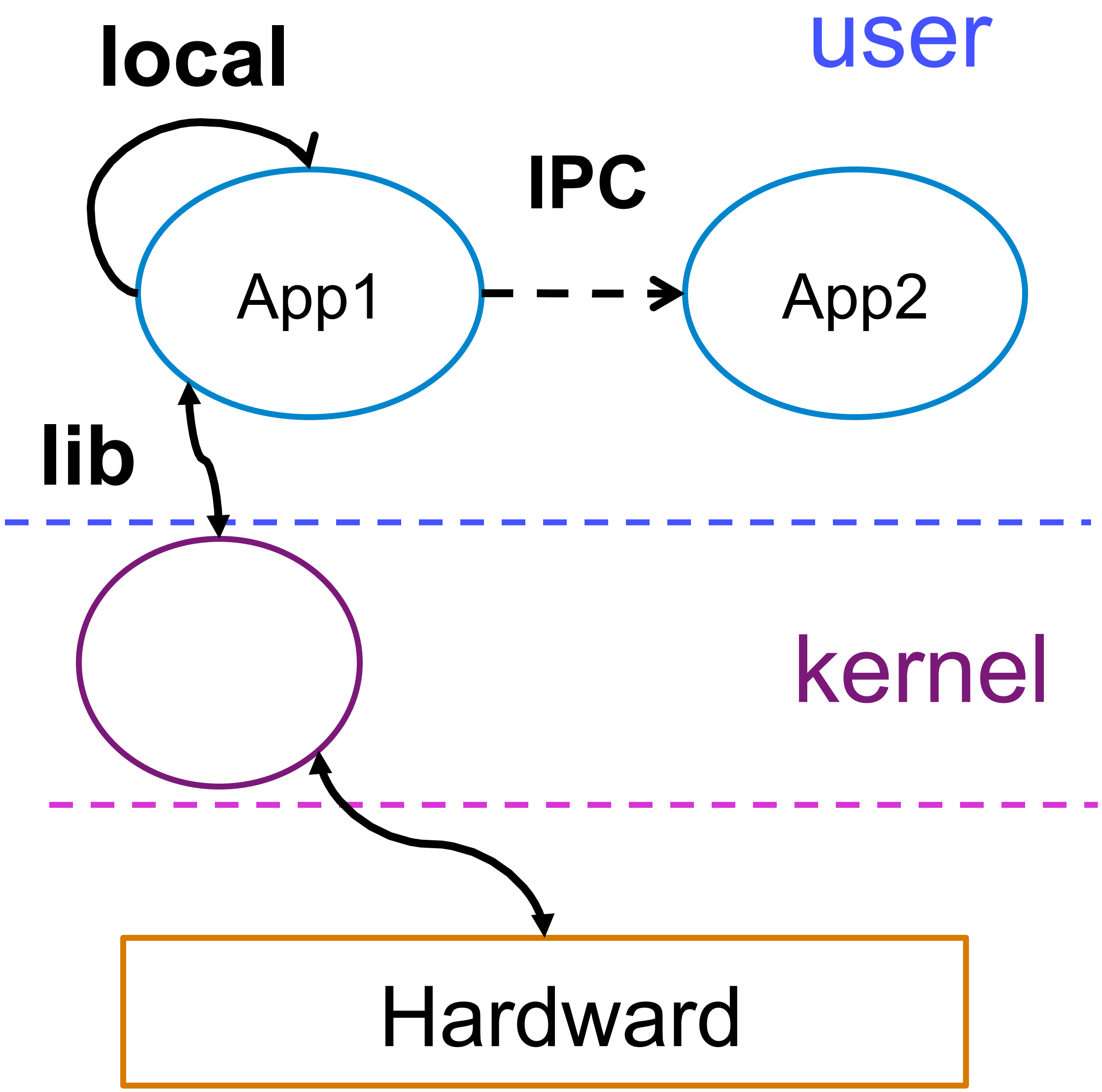
Leading to a picture like this

Within a single processor

A function call can be

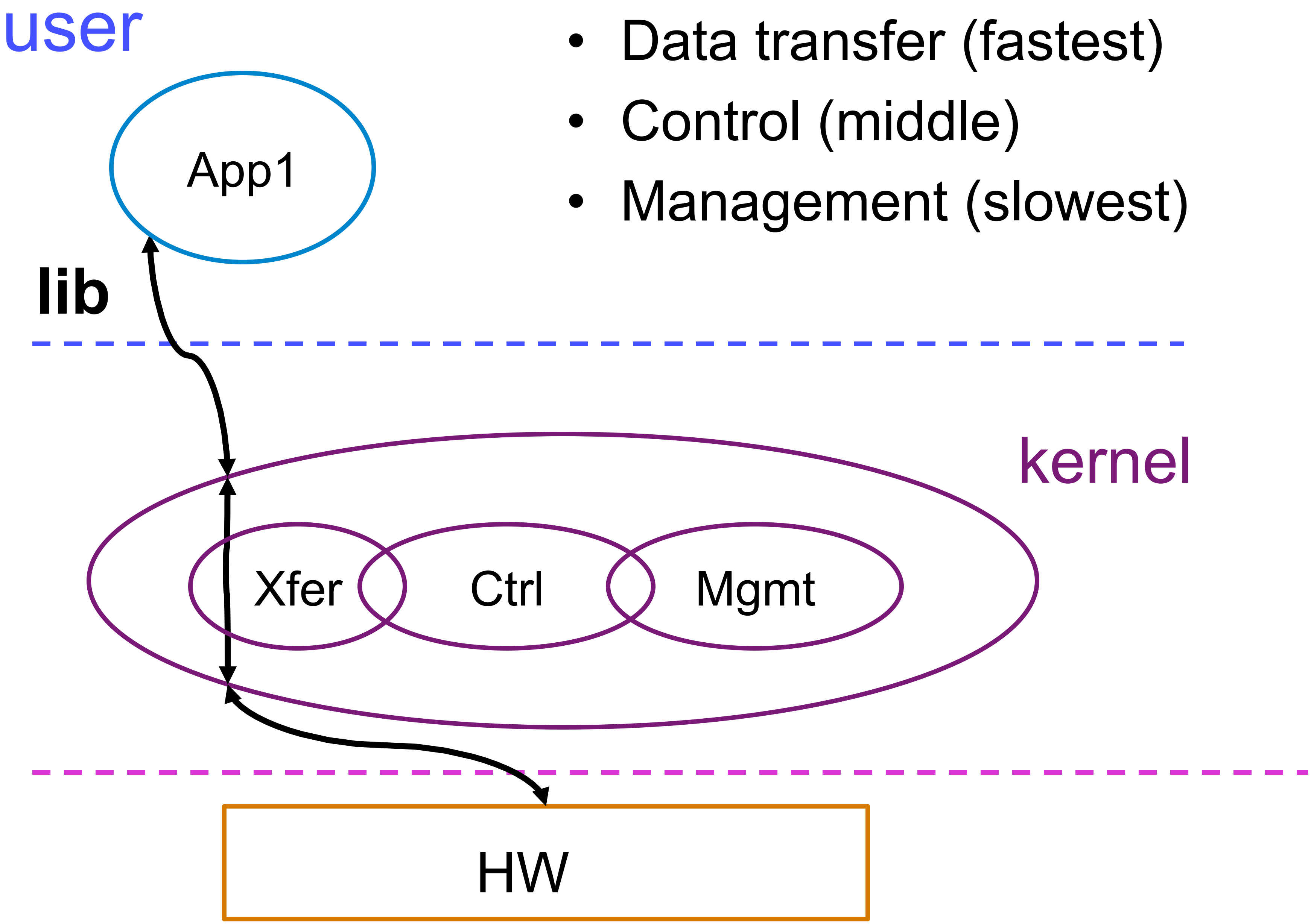
- Local
- Library (system)
- IPC

IPC= InterProcess Communication



The kernel functions are

- Data transfer (fastest)
- Control (middle)
- Management (slowest)



user

App1

lib

kernel

Xfer

Ctrl

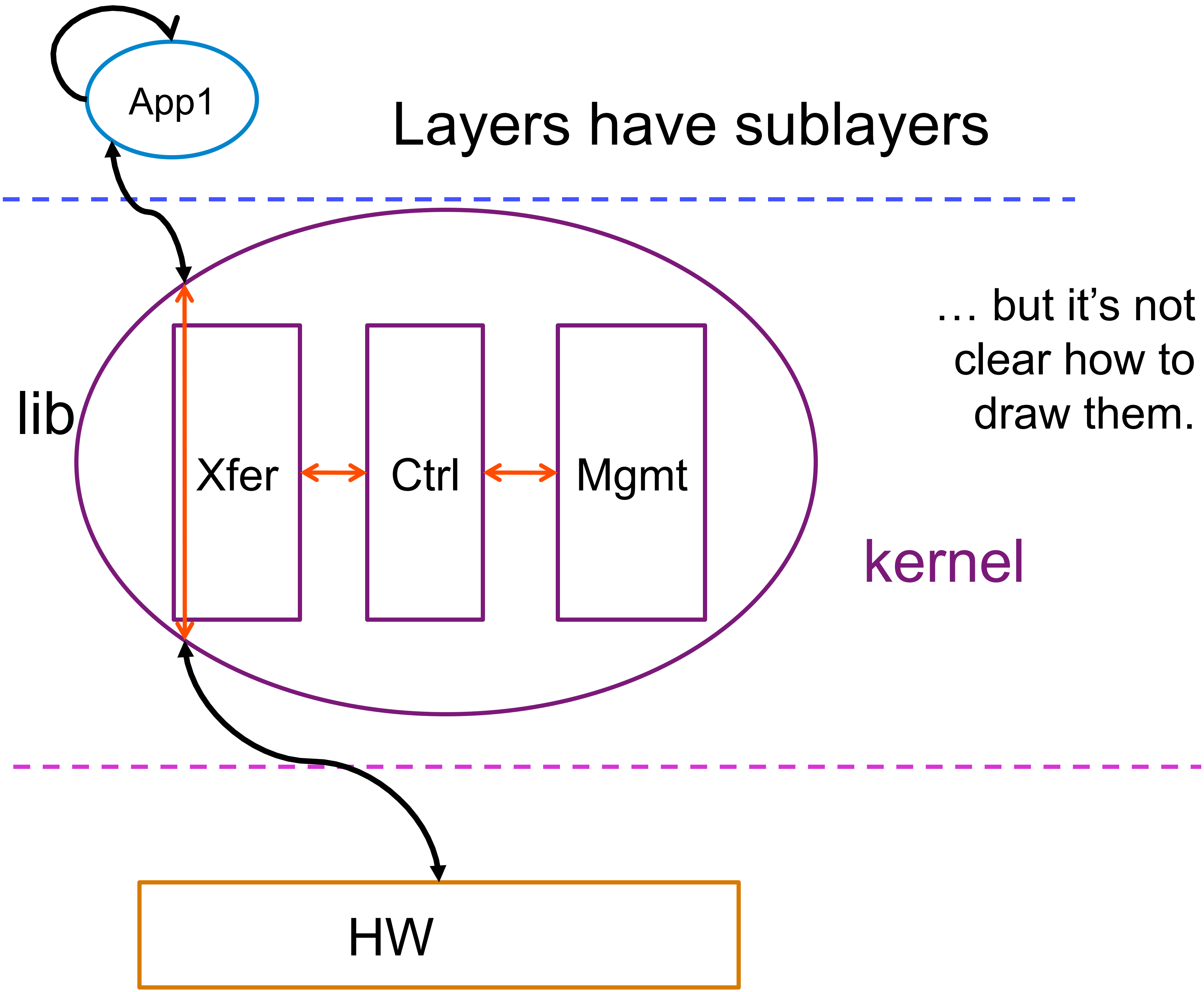
Mgmt

HW

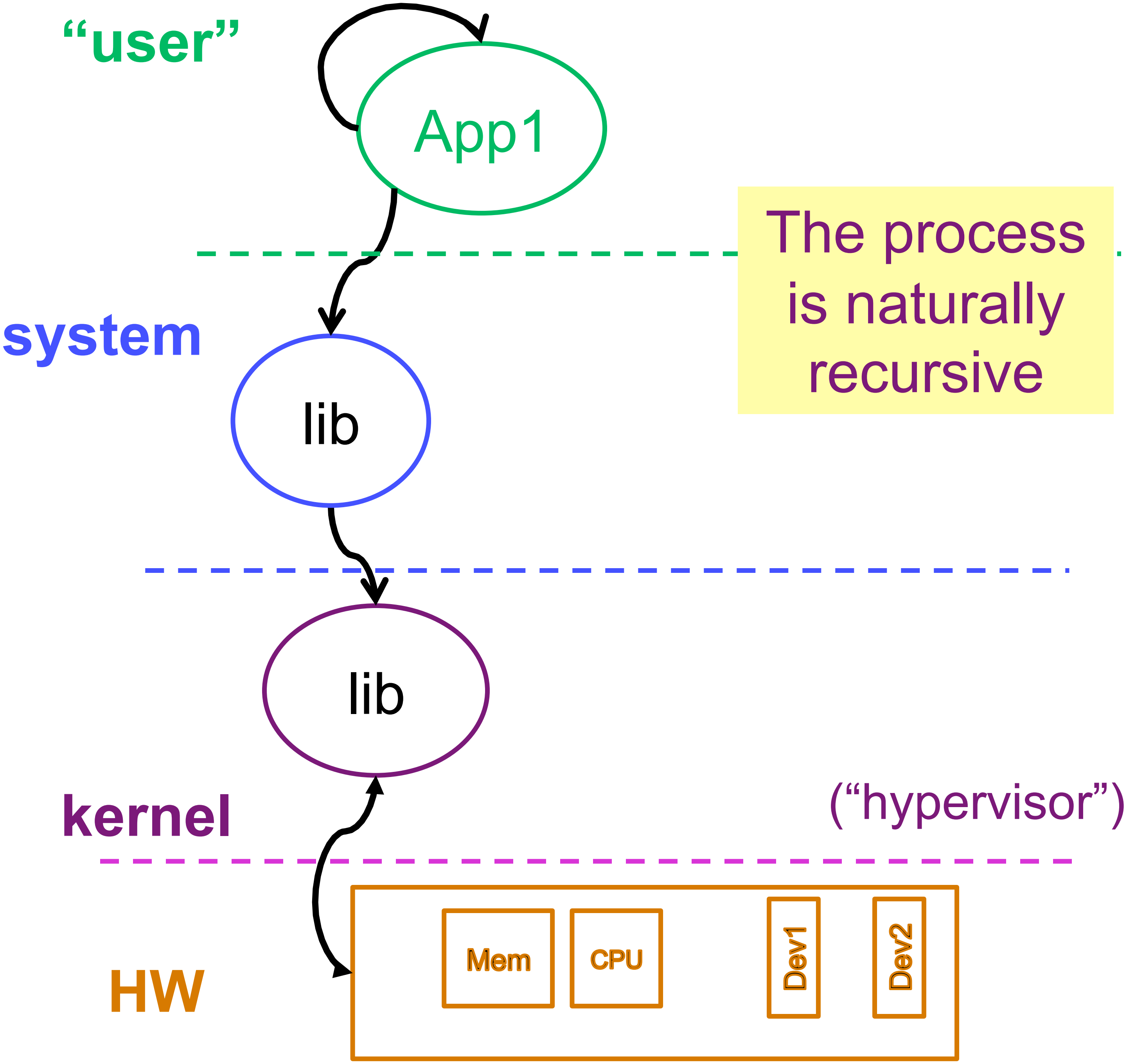
The kernel functions are

- Data transfer (fastest time scale)
  - Within memory (and memory hierarchies)
  - Between devices and memory
  - Between memory and computing elements
- Control (middle time scales)
  - Scheduling/Multiplexing resources
  - In time and space
- Management (slowest time scale)
  - ***What*** resources are available?
  - ***Where*** are they?

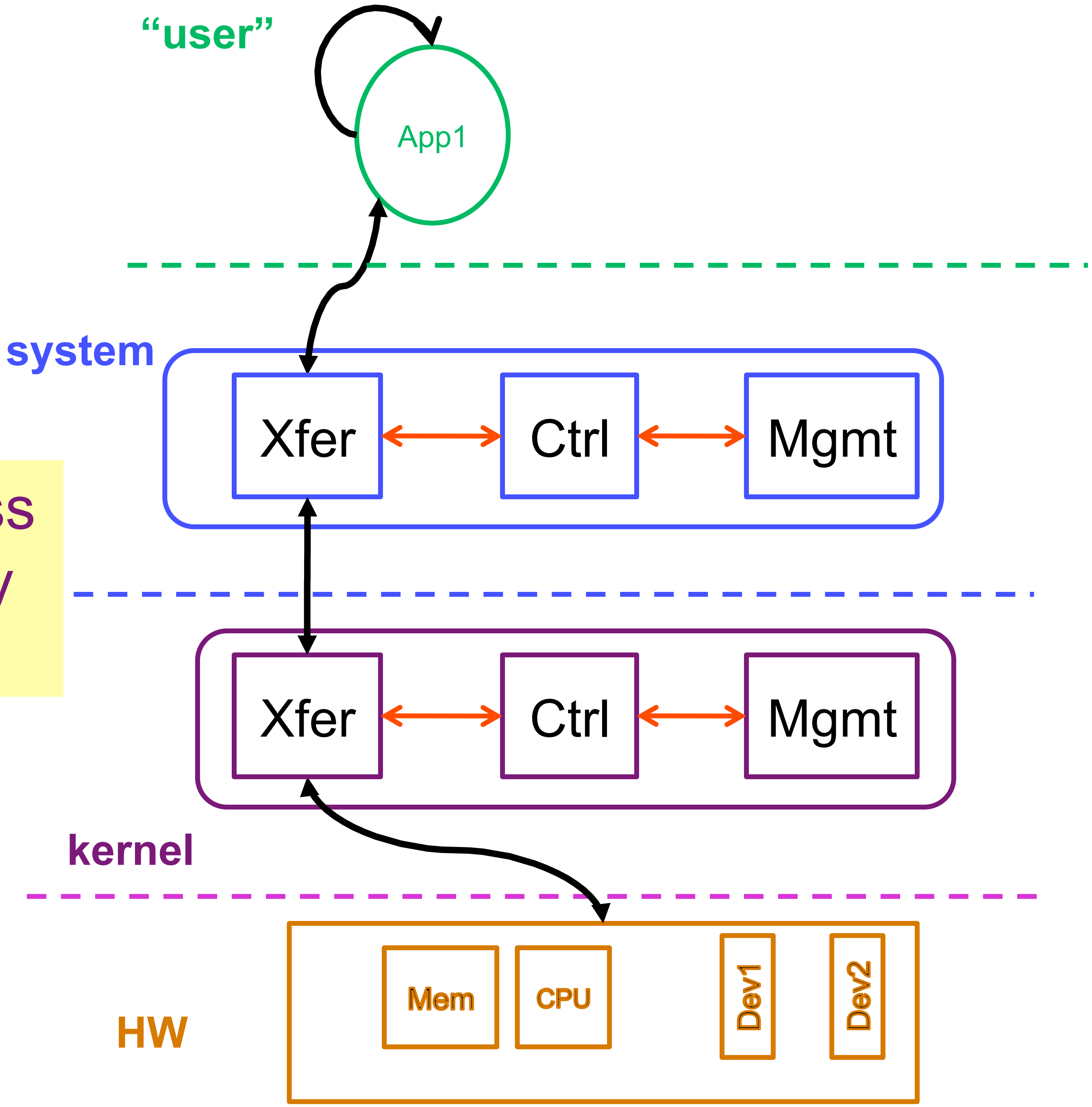
Layers have sublayers



… but it’s not clear how to draw them.

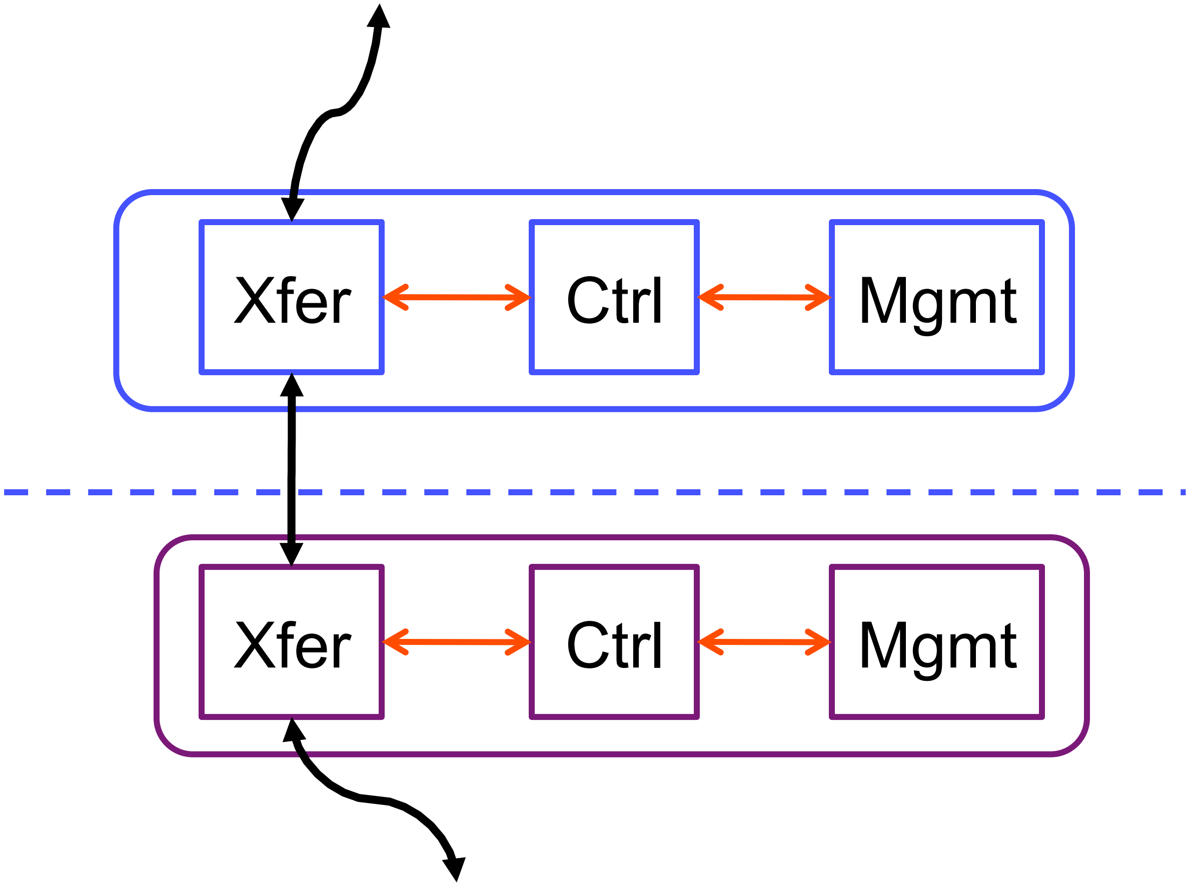


The process is naturally recursive

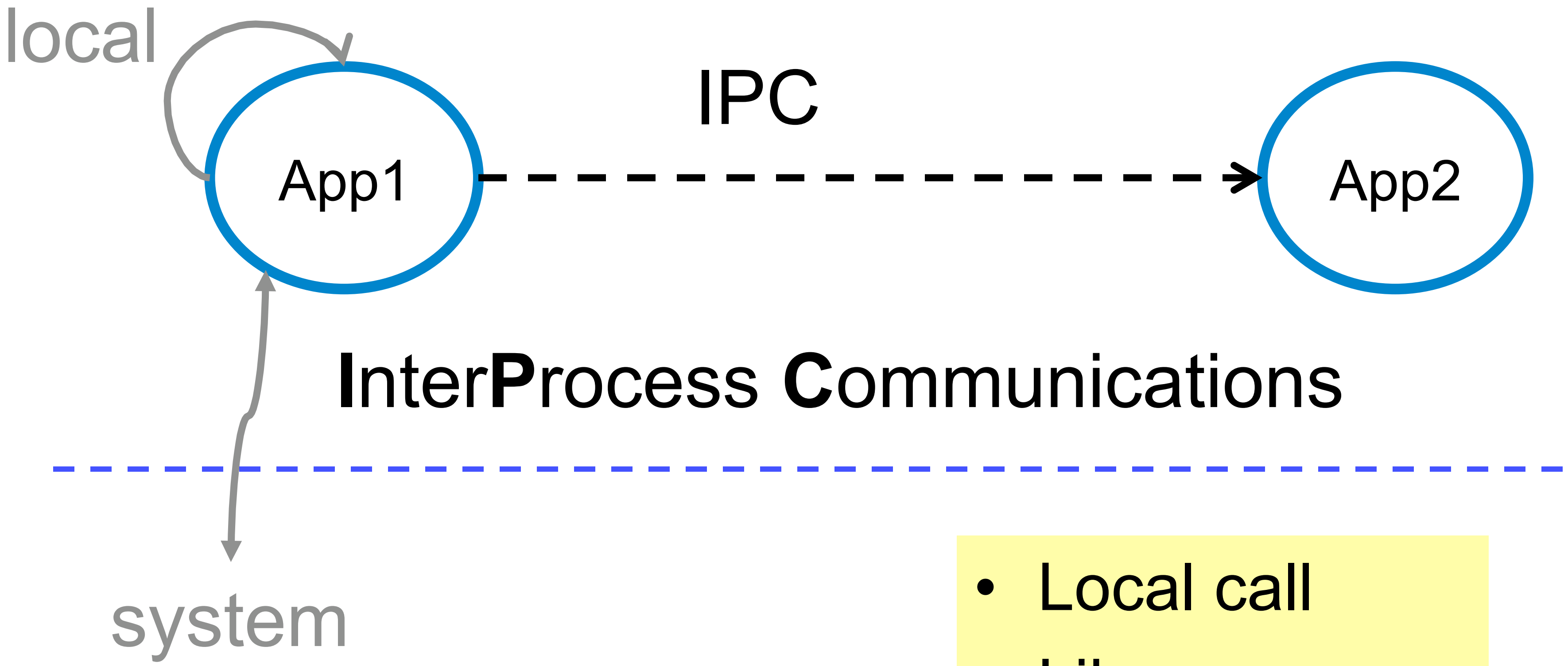


The process is naturally recursive

Layers have sublayers



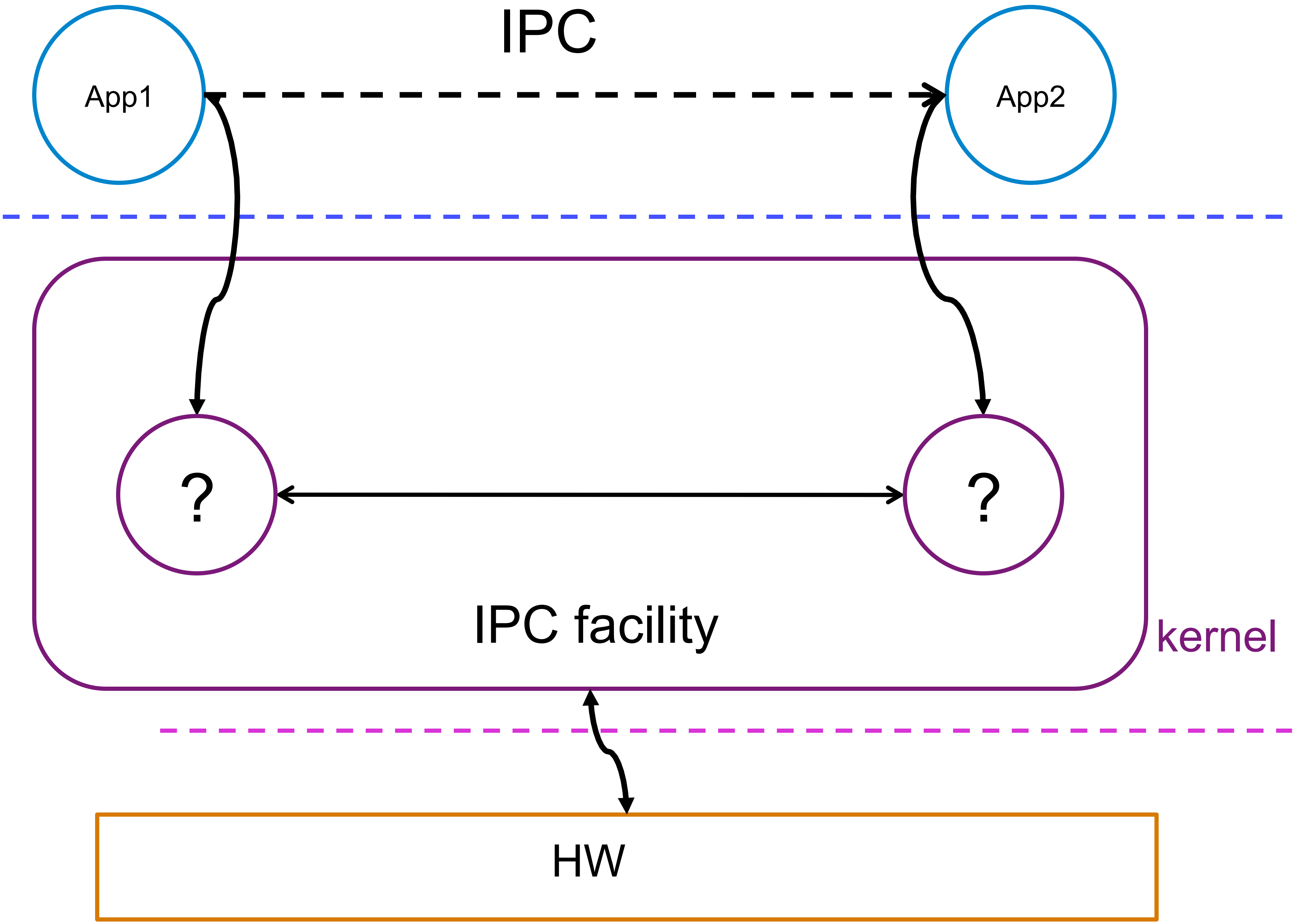
Layers are naturally recursive

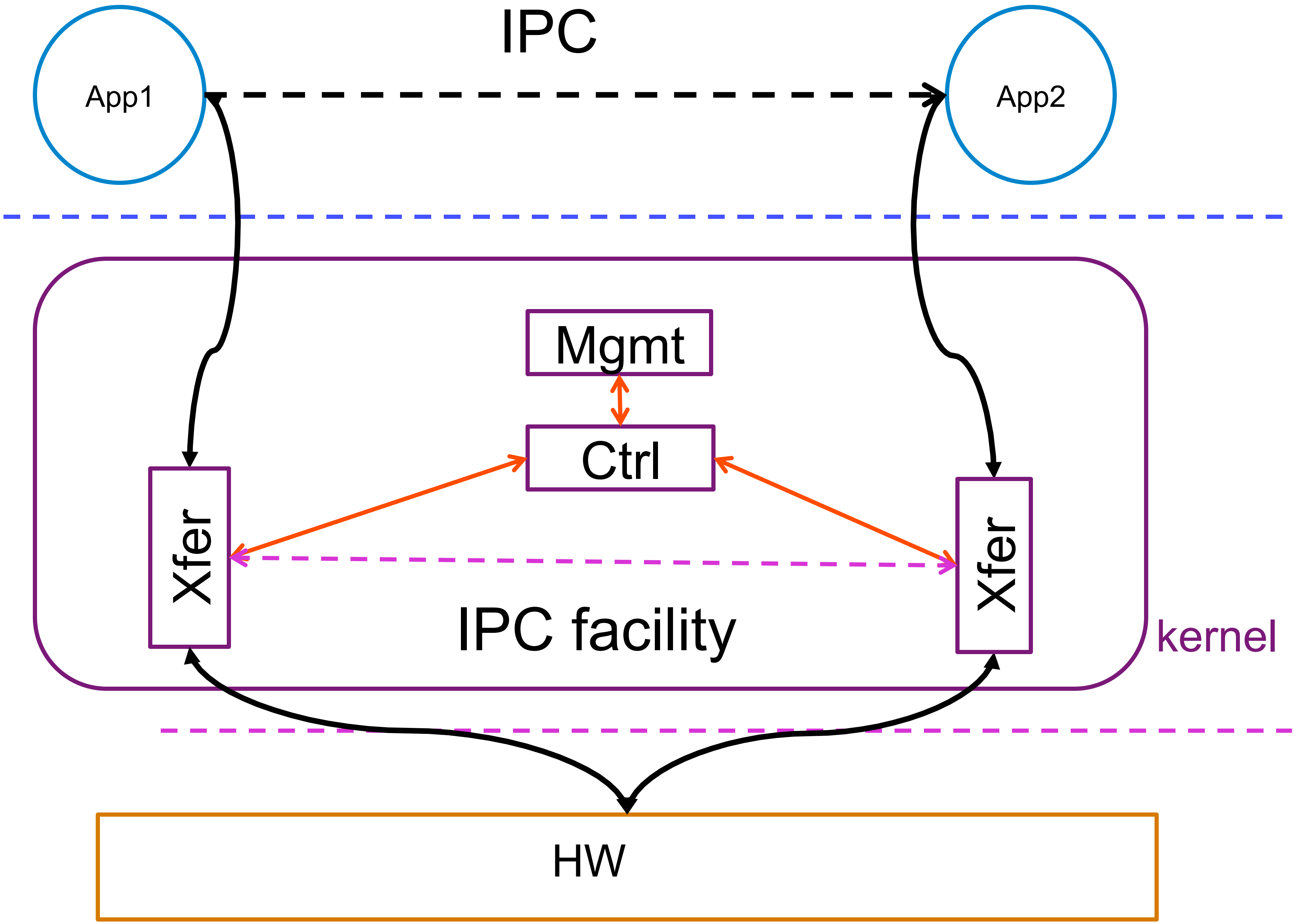


Inter**P**rocess **C**ommunications

- Local call
- Library (system) call
- IPC

Want them all to behave similarly.


IPC
App1
App2
?
?
IPC facility
kernel
HW



IPC

App1

App2

Mgmt

Ctrl

Xfer

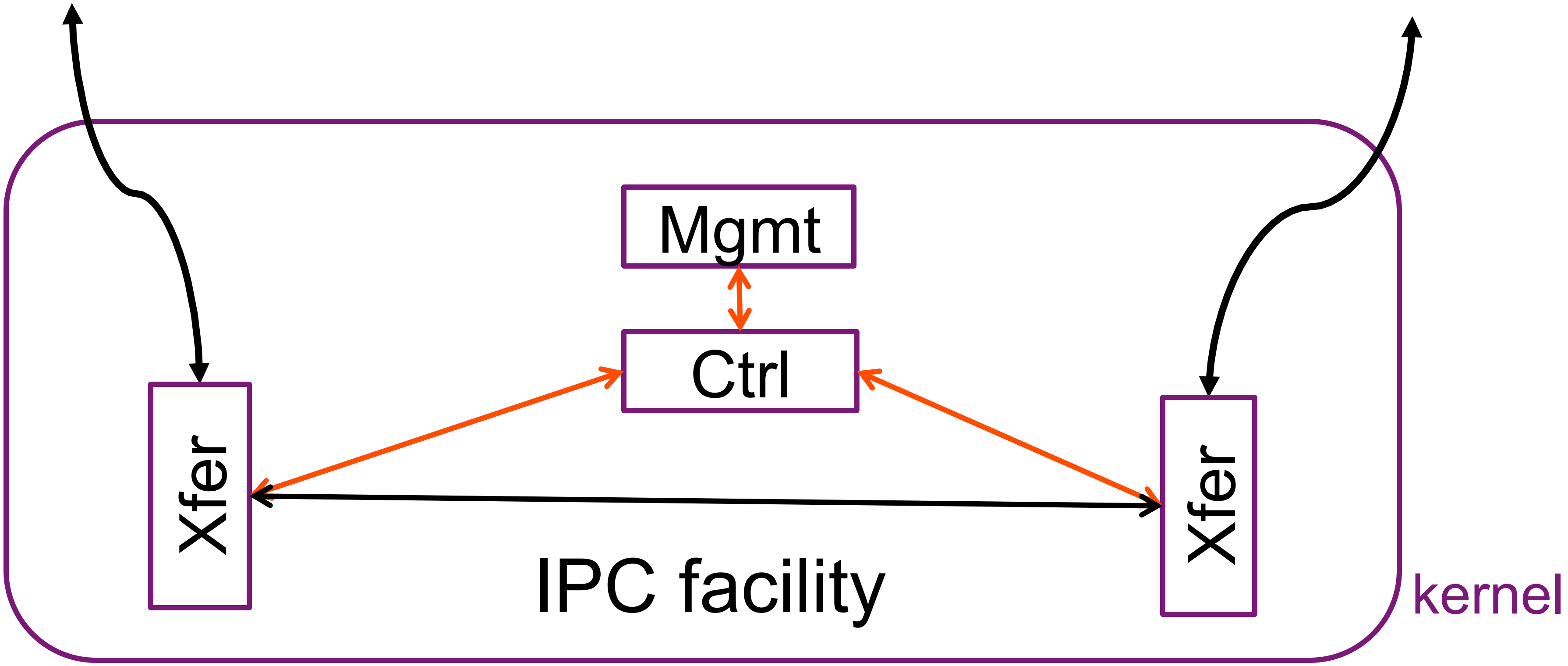
Xfer

IPC facility

kernel

HW

Layers have sublayers



Mgmt

Ctrl

Xfer

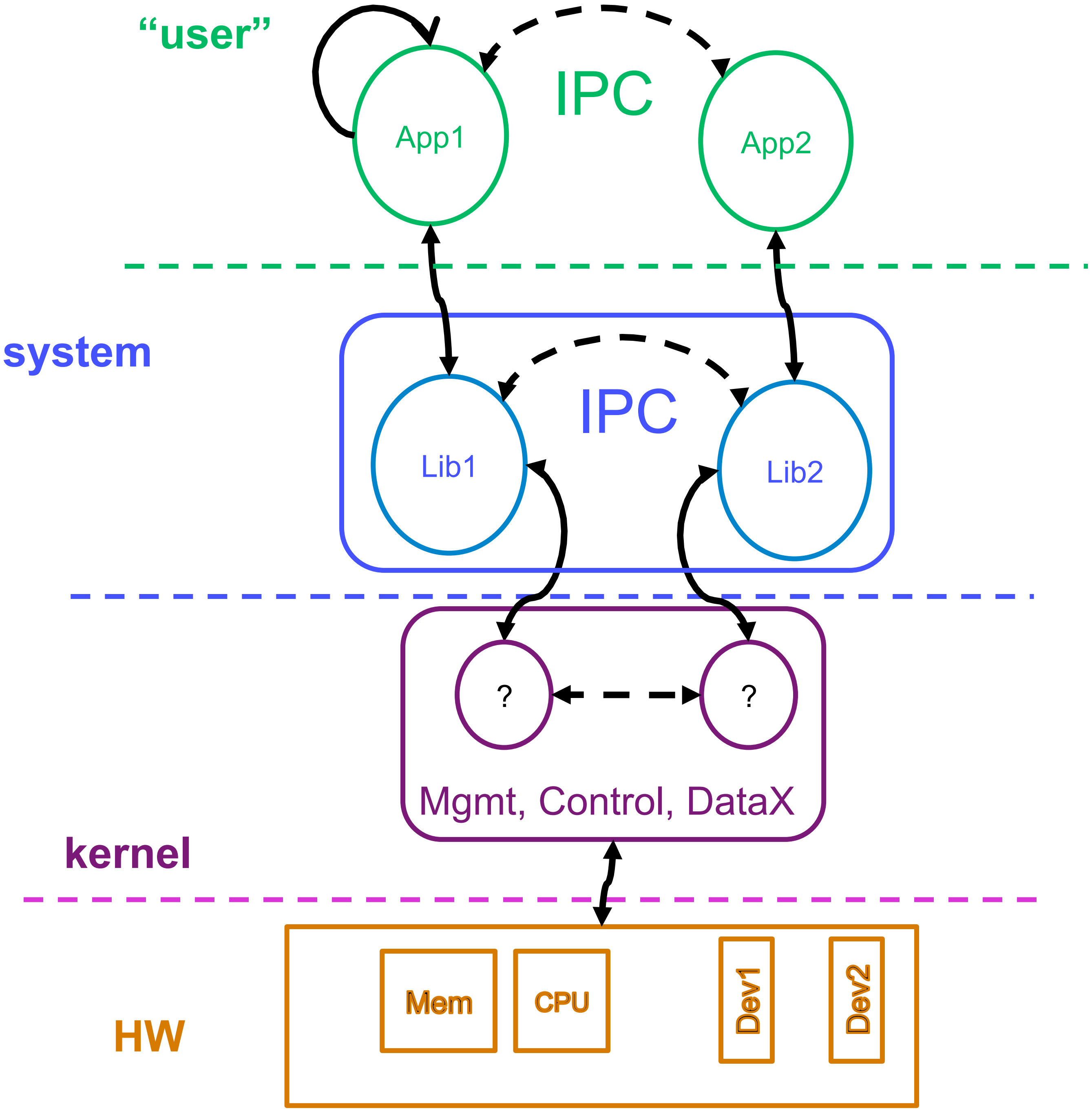
Xfer

IPC facility

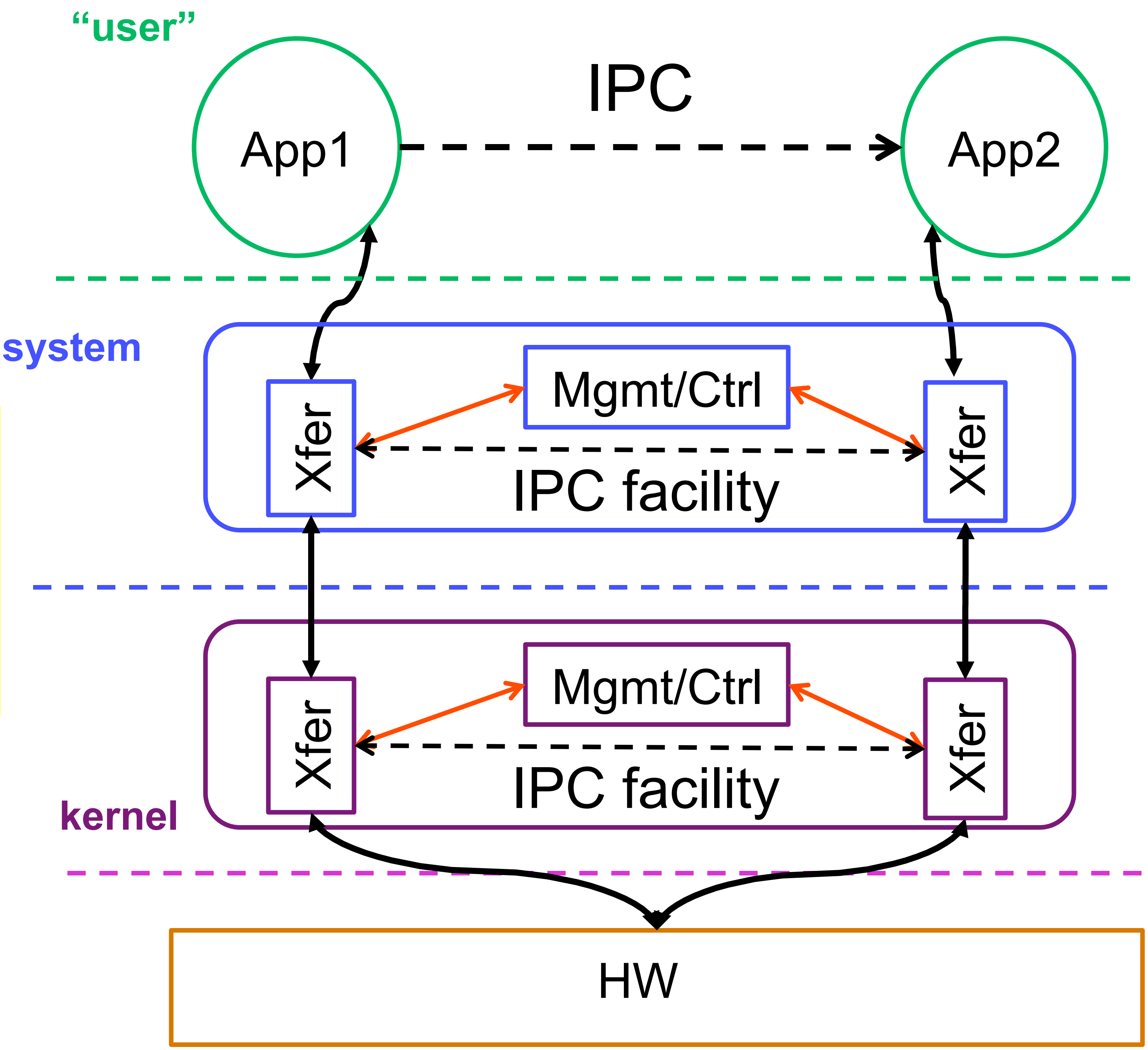
kernel

… but it's not clear how to draw them.

IPC is naturally recursive



IPC is naturally recursive



"user"

App1

IPC

App2

system

Mgmt/Ctrl

Xfer

Xfer

IPC facility

Mgmt/Ctrl

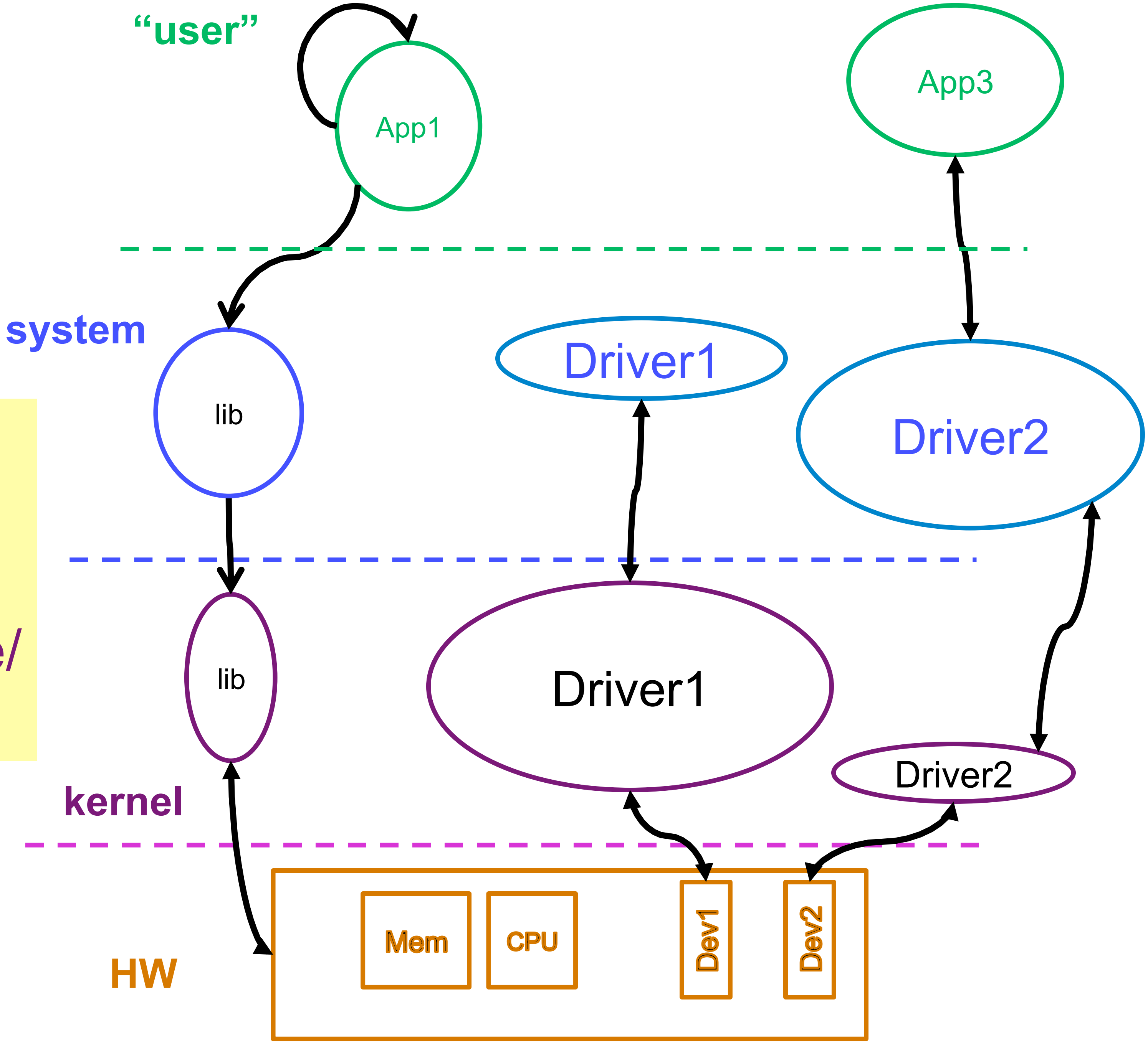
Xfer

Xfer

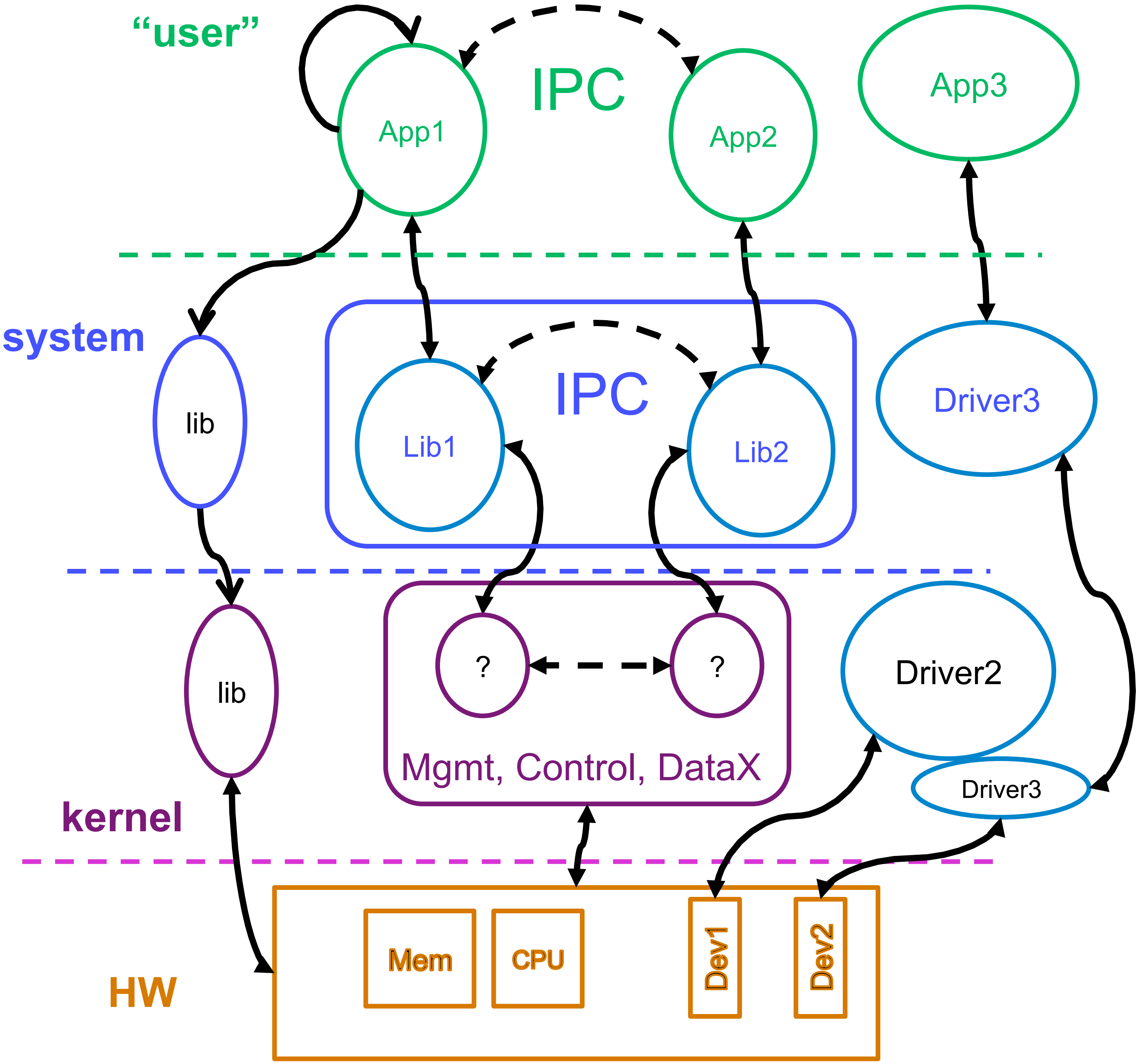
IPC facility

kernel

HW

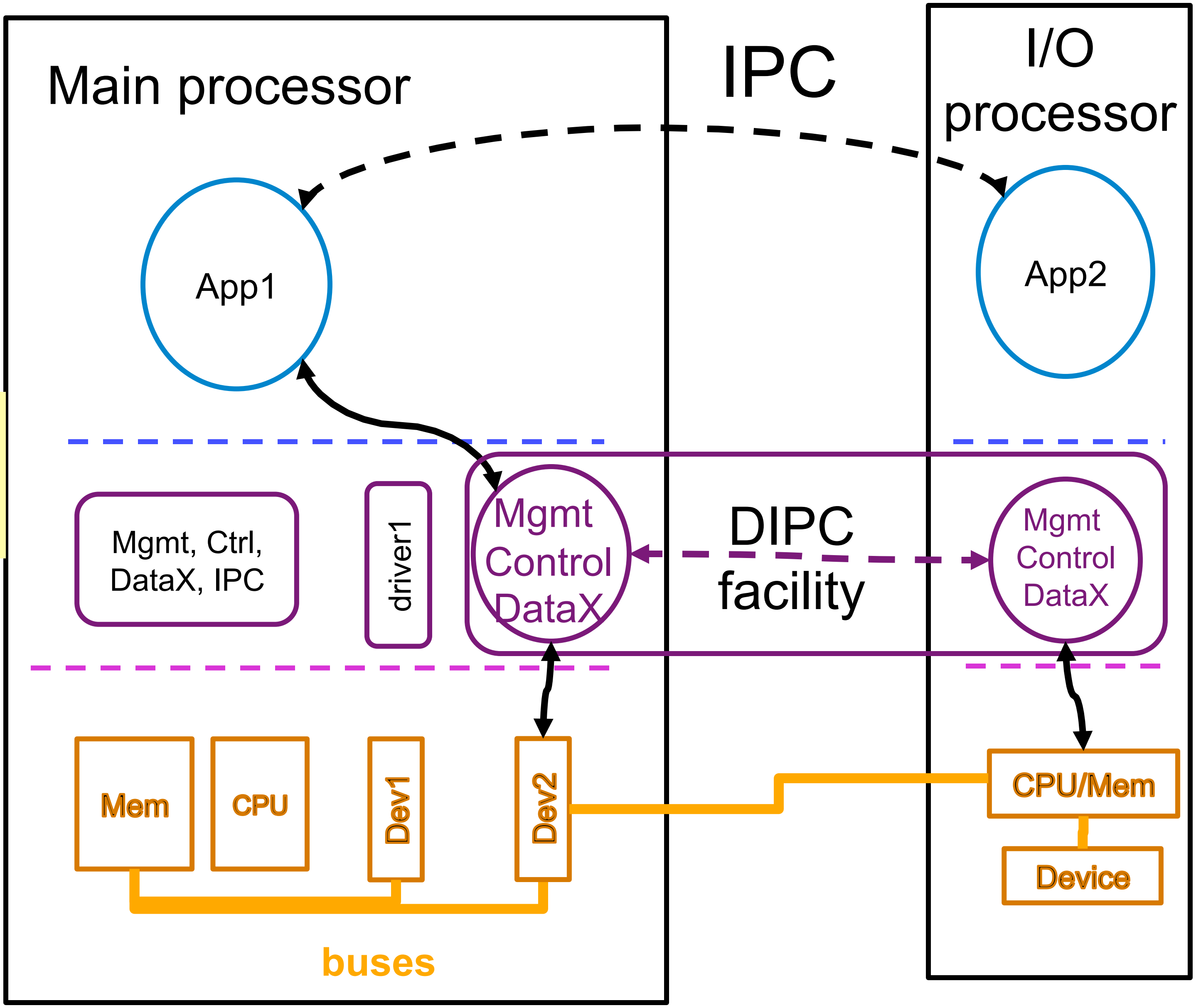


Design choices effect performance/ robustness



layers are naturally recursive

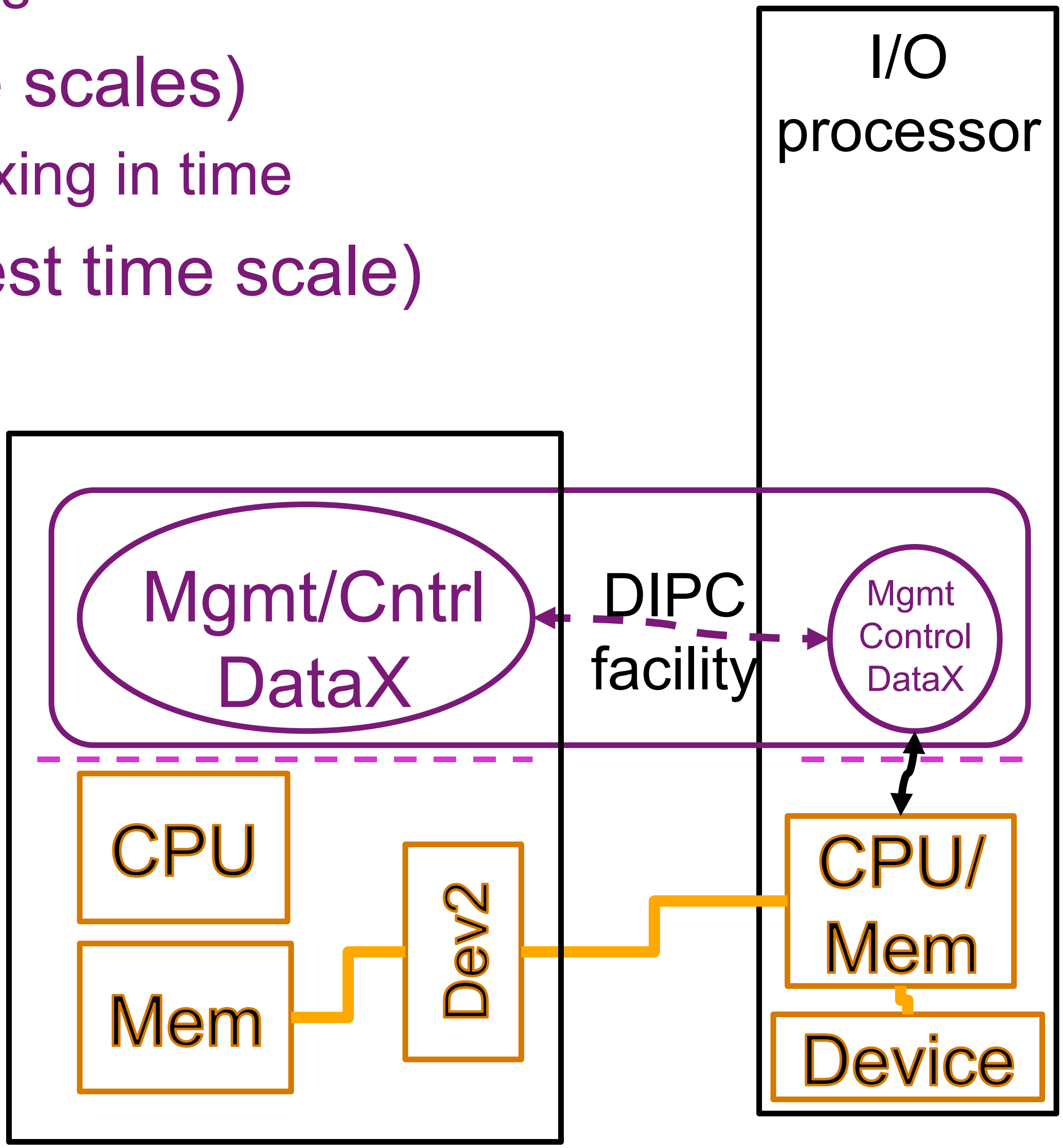
What happens in a computer ***system***?



Distributed
IPC.

- Data transfer (fastest time scale)
  - Between "processors"
- Control (middle time scales)
  - Scheduling/Multiplexing in time
- Management (slowest time scale)
  - What? Where?

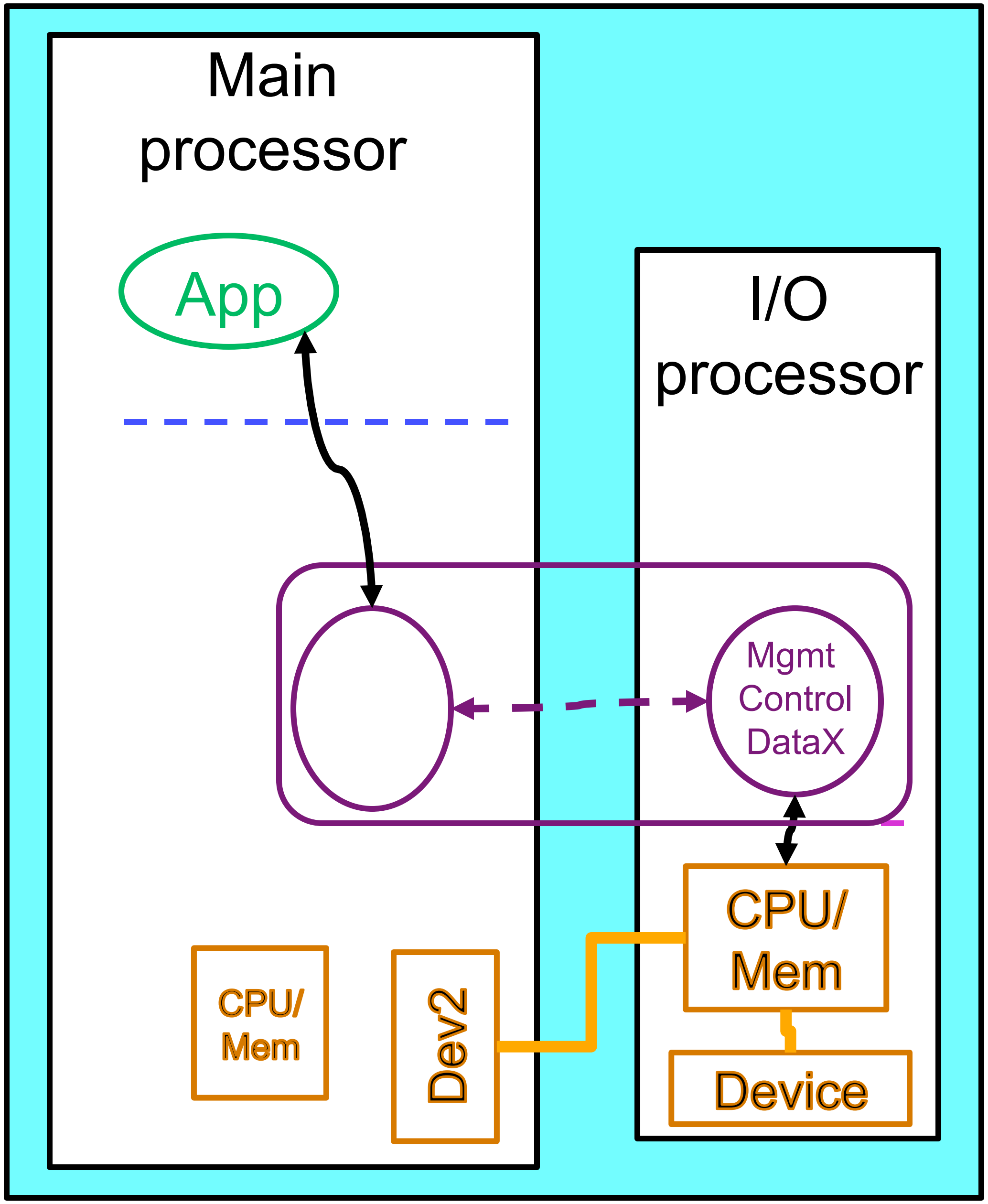
Mgmt and Ctrl become more complex



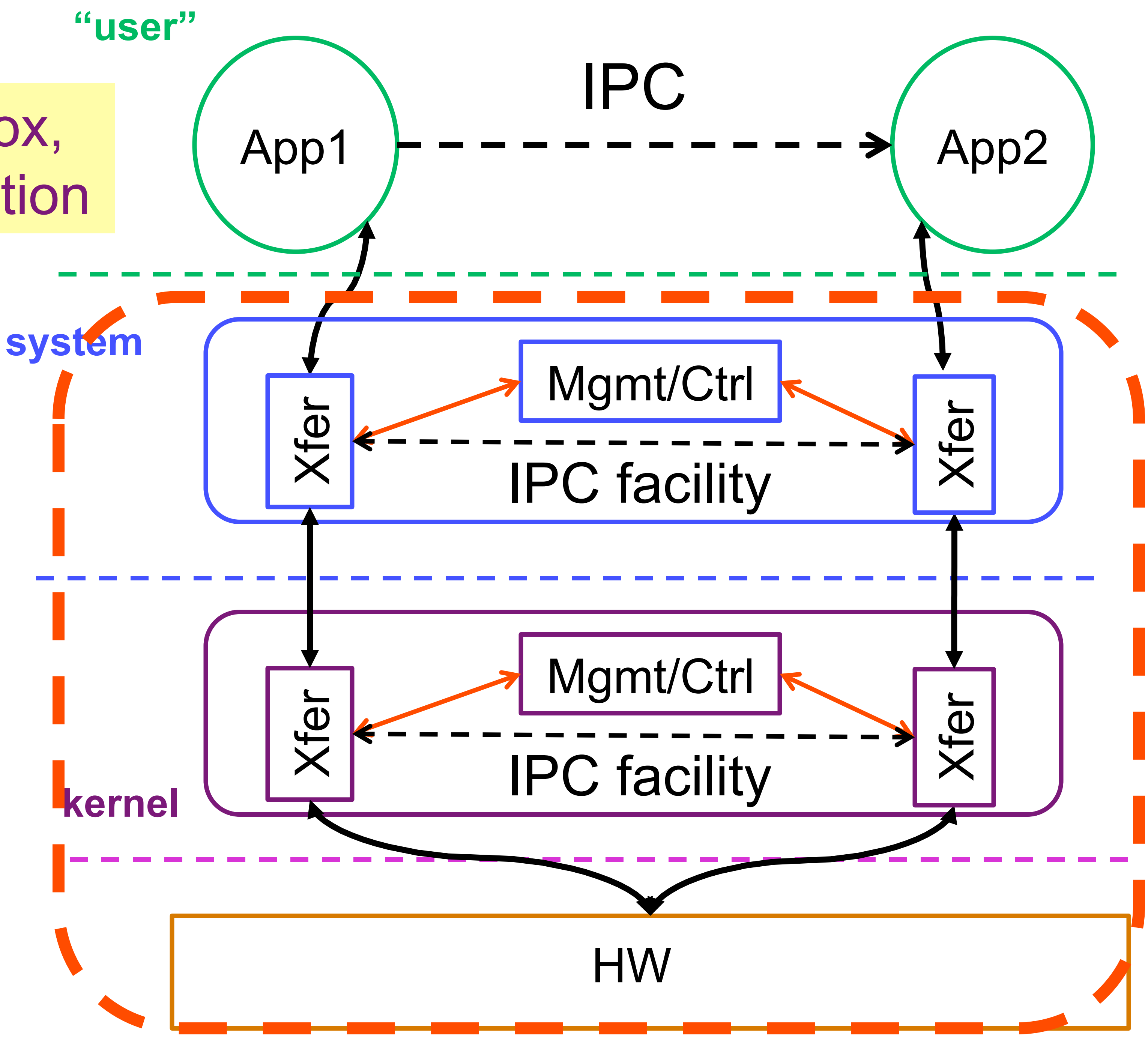
Any layer's functions are

- Data transfer (fastest time scale)
  - Within memory (and memory hierarchies)
  - Between devices and memory
  - Between memory and computing elements
  - Between virtualized resources (in higher layers)
- Control (middle time scales)
  - Scheduling/Multiplexing resources
  - In time and space
  - Real and virtualized
- Management (slowest time scale)
  - ***What*** resources are available?
  - ***Where*** are they?

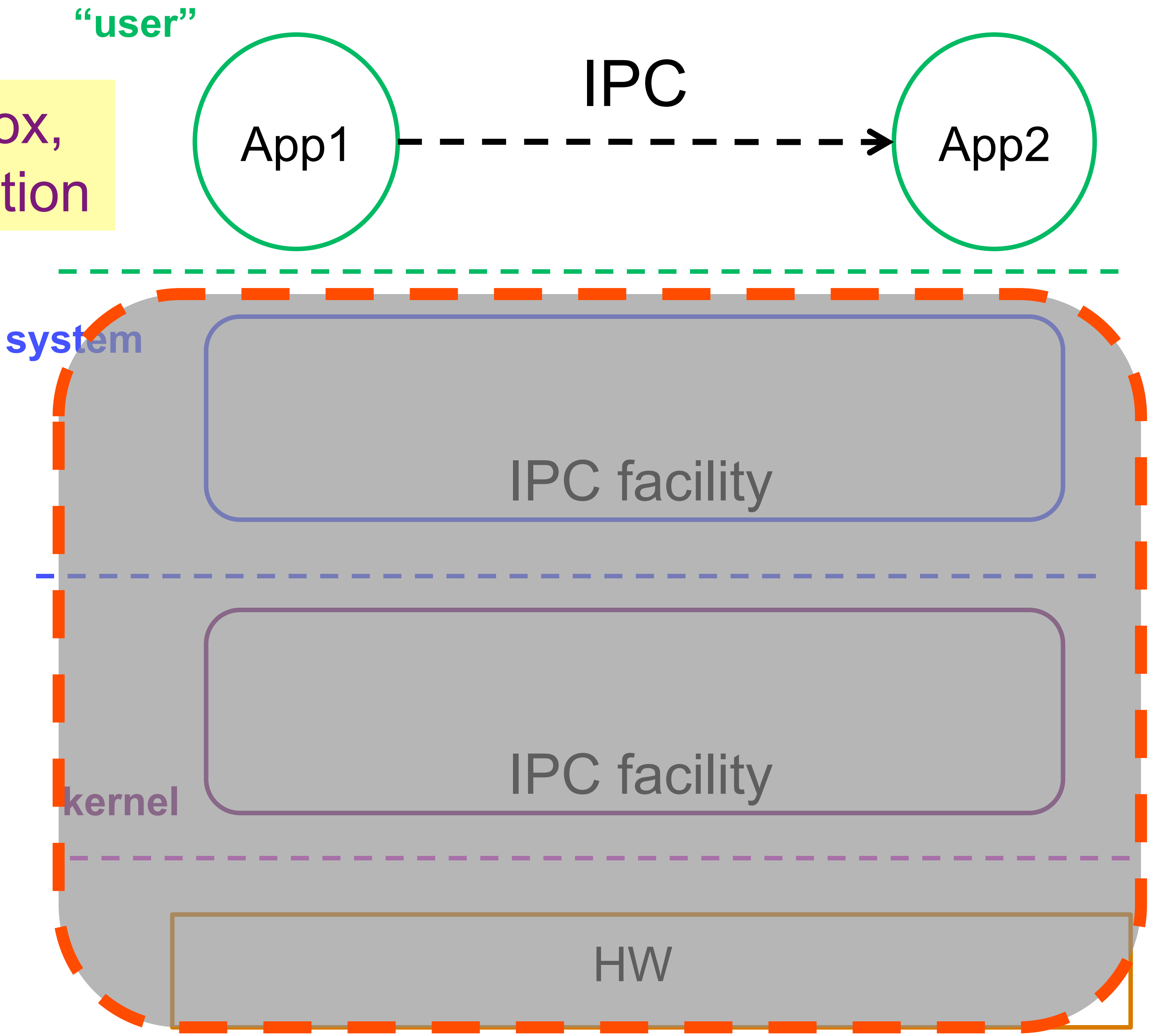
Might be all in the same “box”.



Black box, virtualization



Black box,
virtualization



"user"

IPC

App1

App2

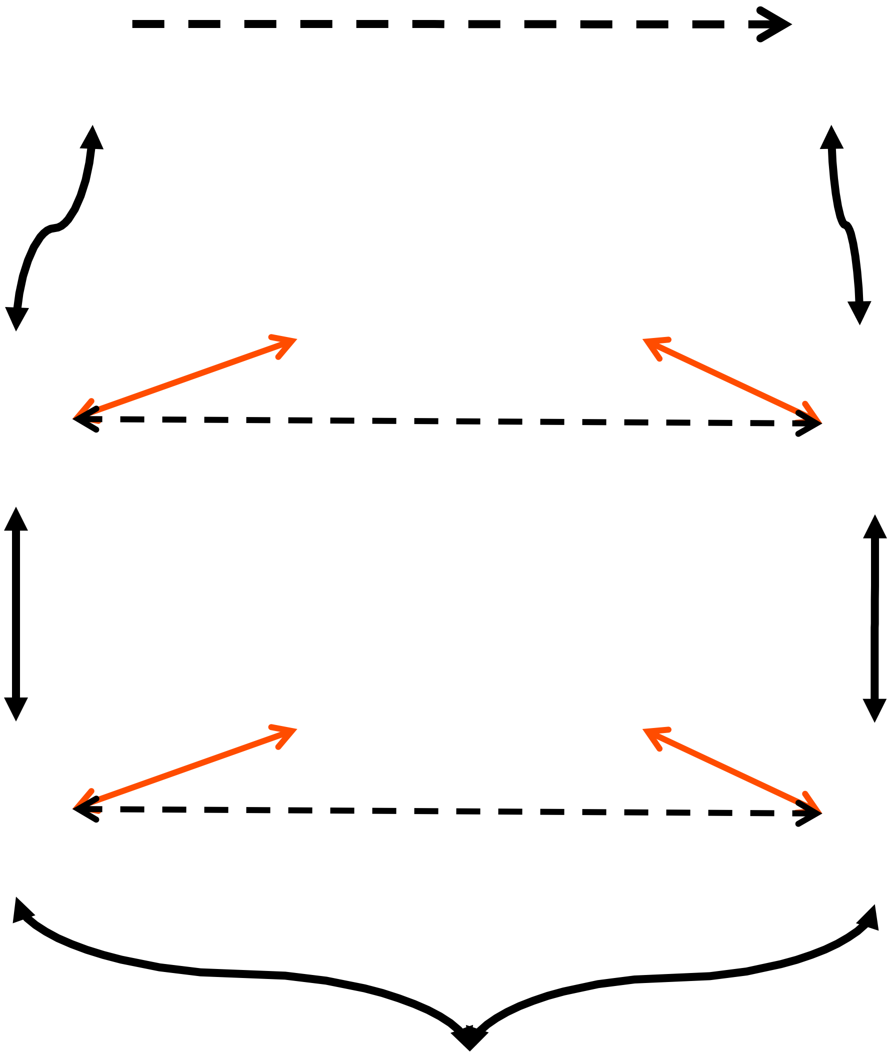
system

IPC facility

kernel

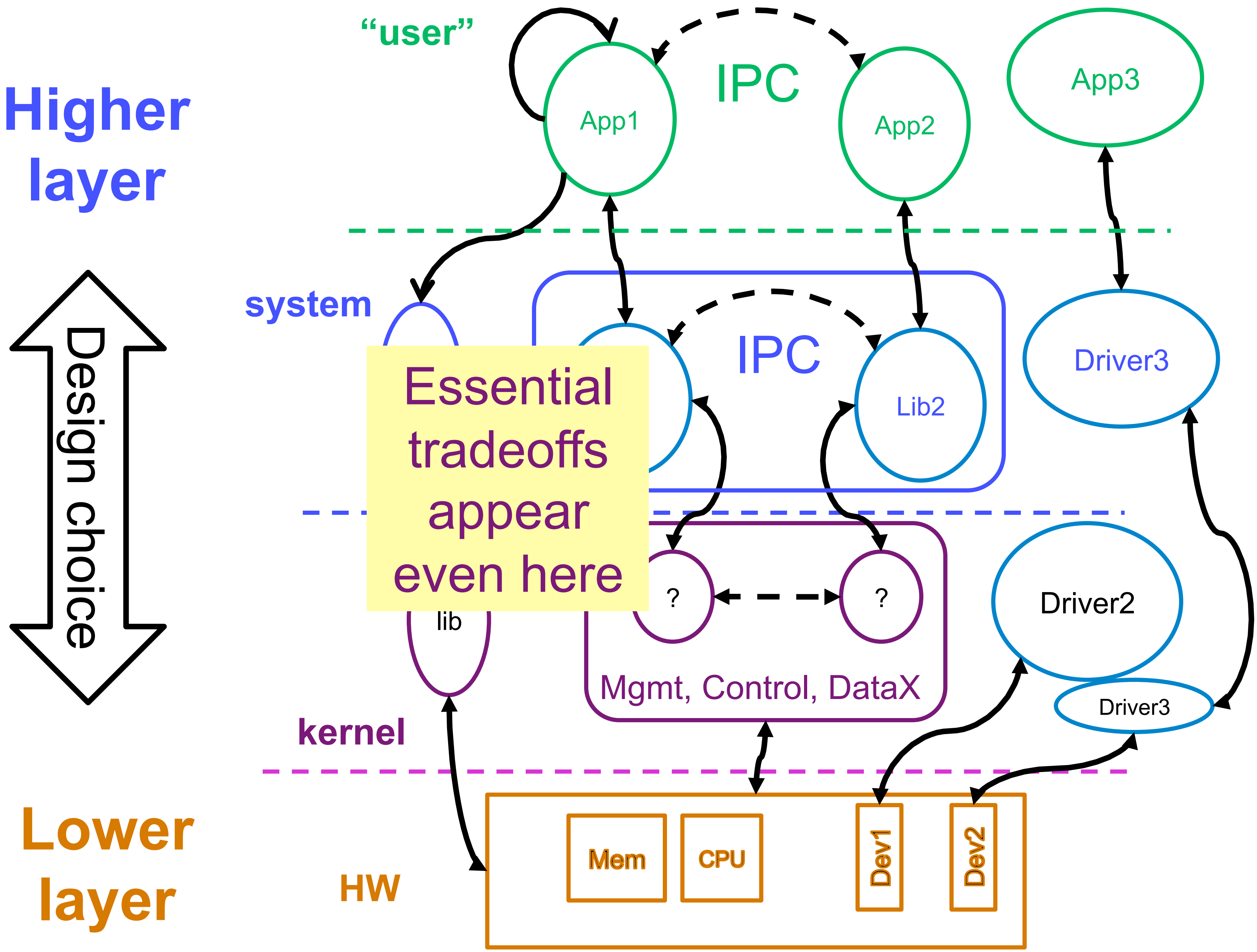
IPC facility

HW



All these signals are "virtual"

The only "real" signals are not shown



Essential tradeoffs appear even here

Higher layer

Design choice

Lower layer

"user"

App1

IPC

App2

App3

system

Lib2

Driver3

lib

?

?

Mgmt, Control, DataX

Driver2

Driver3

kernel

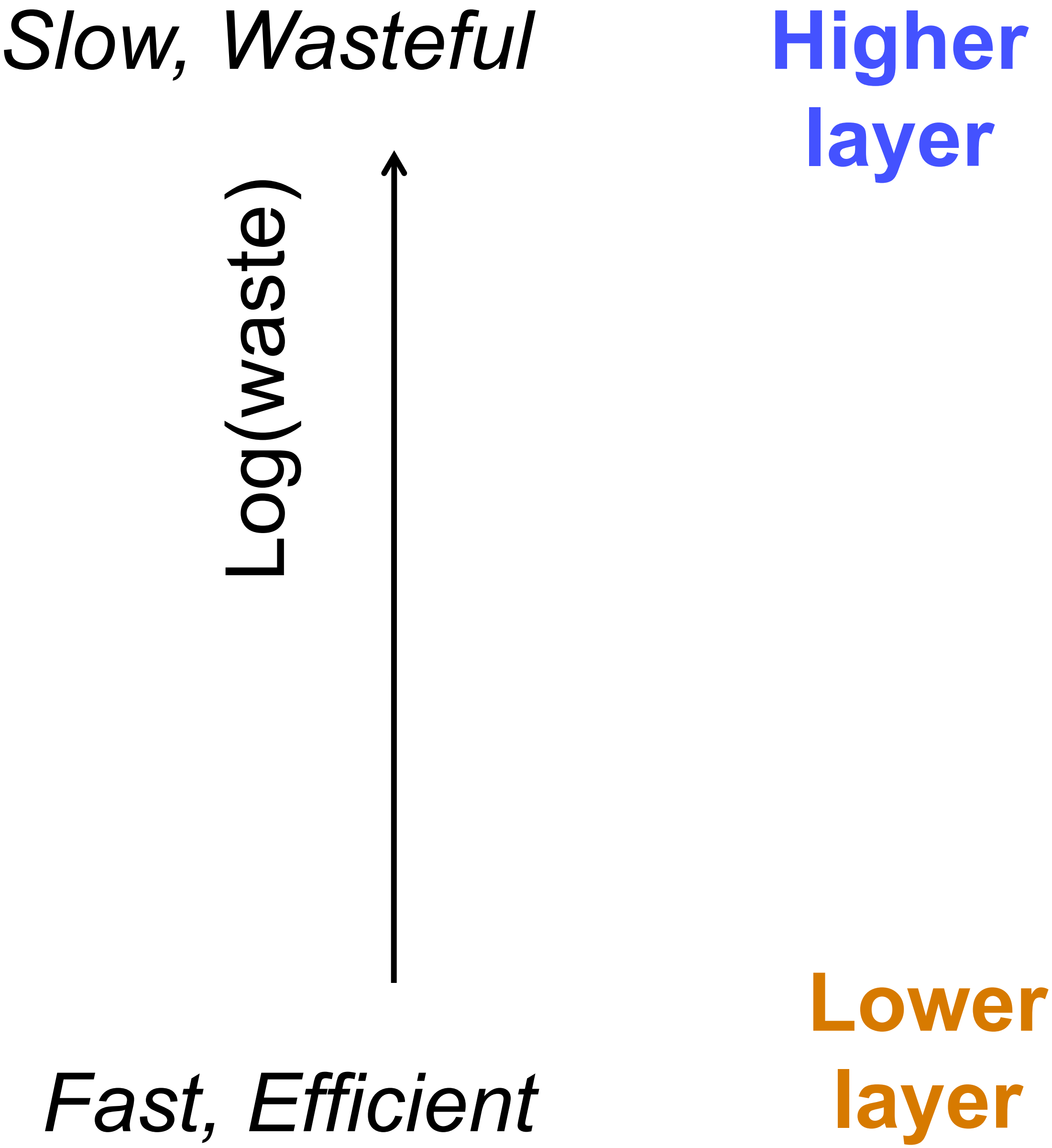
HW

Mem

CPU

Dev1

Dev2



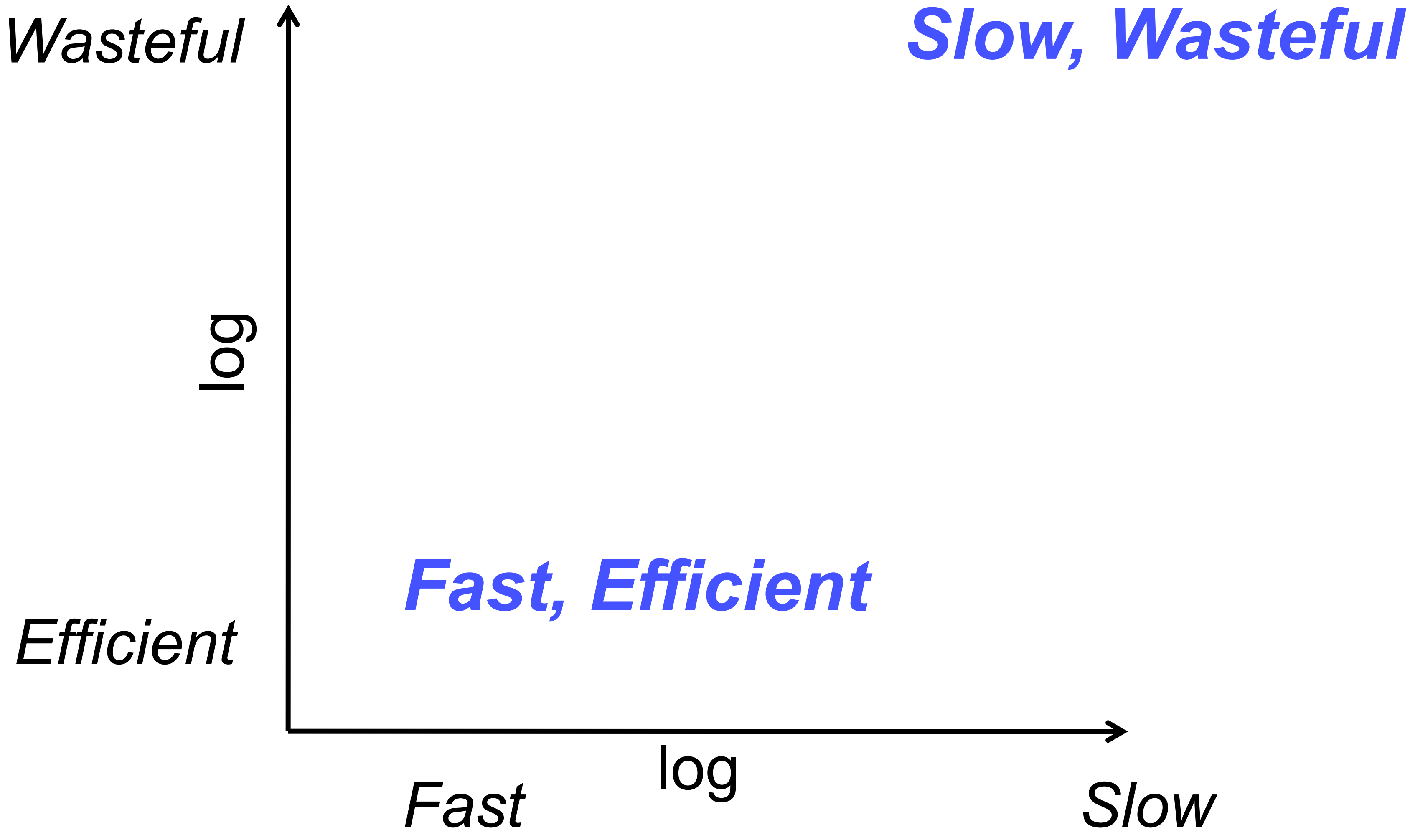
Slow, Wasteful

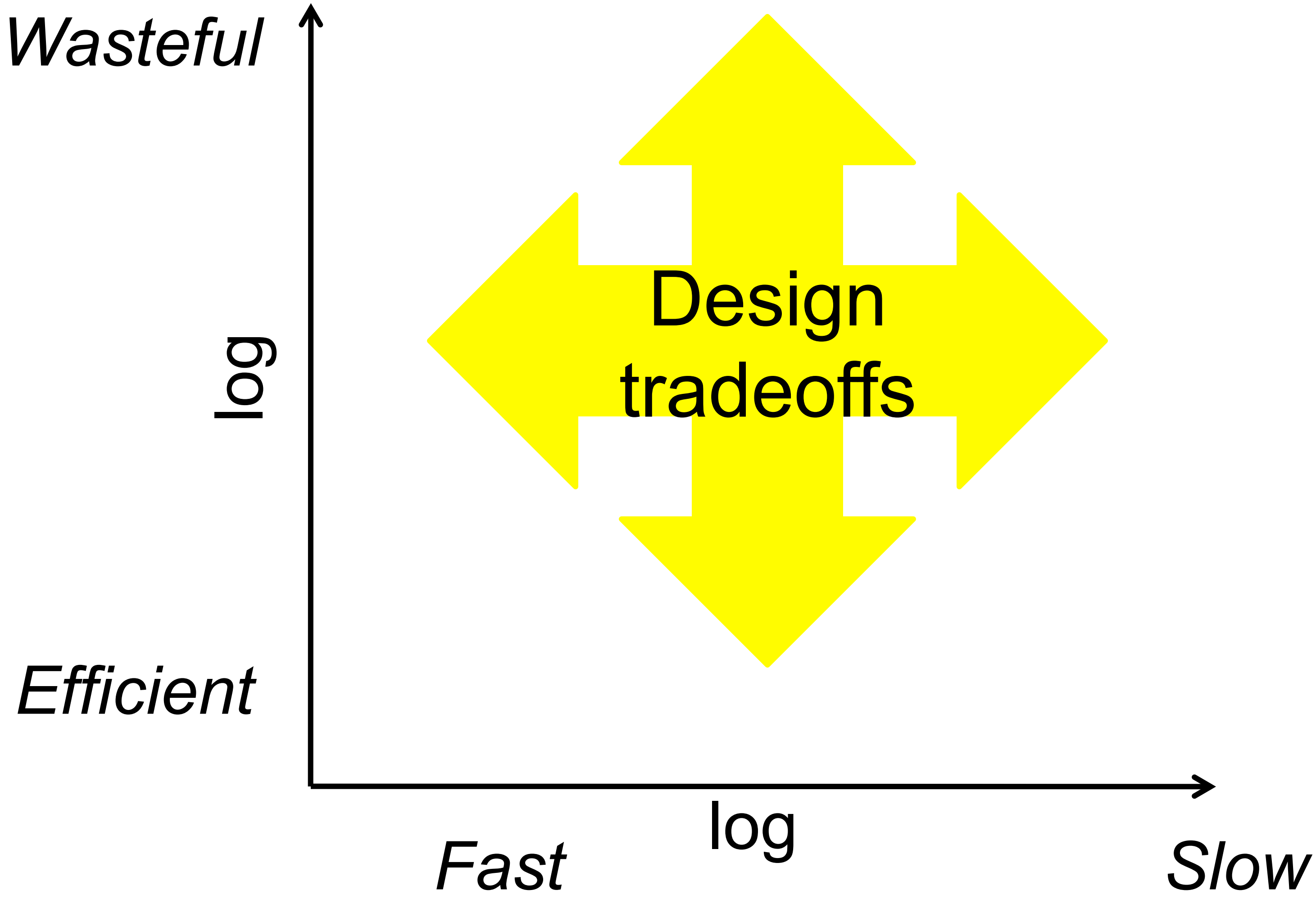
Higher layer

Fast, Efficient

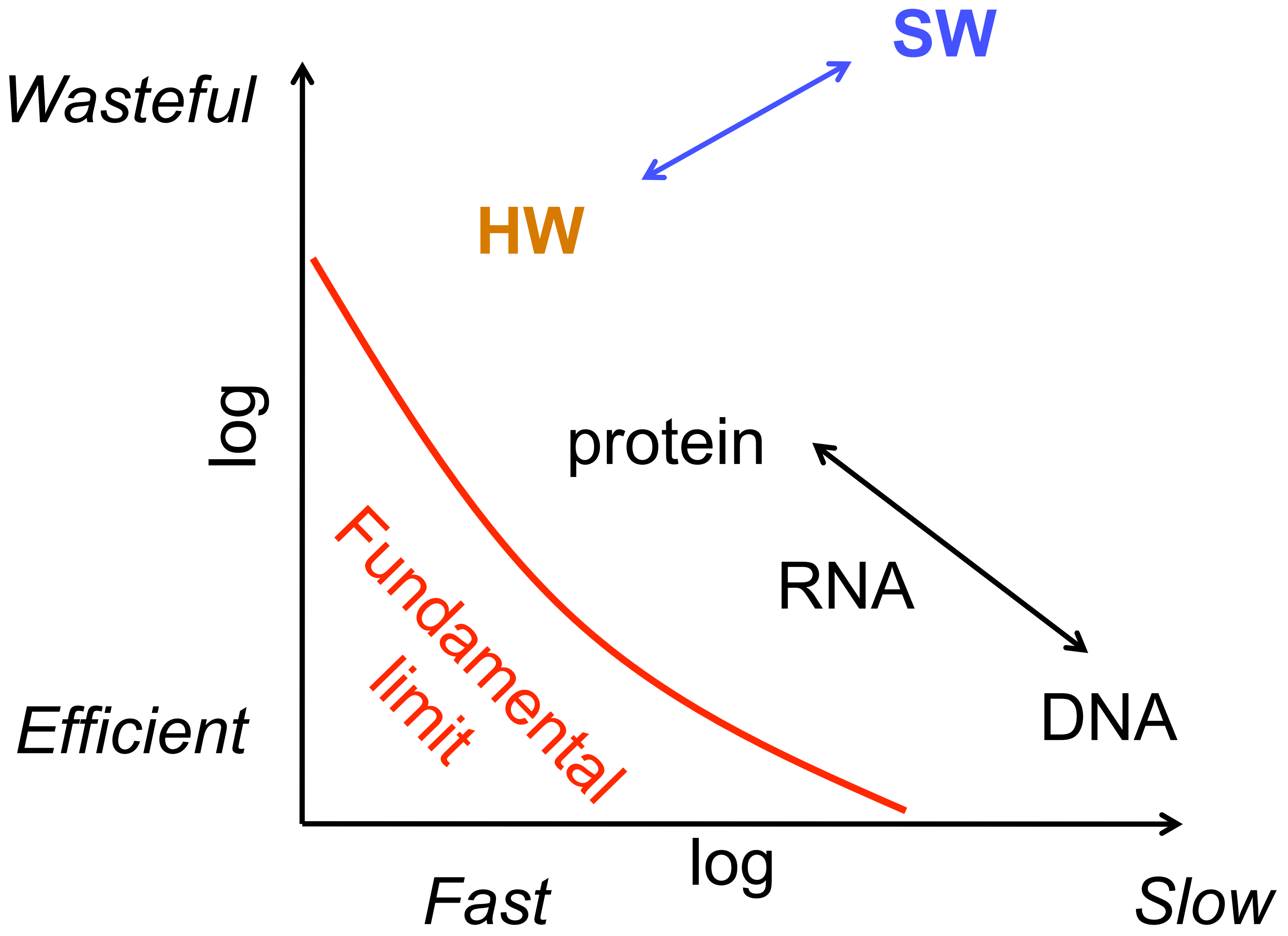
Lower layer

Expand dimensions

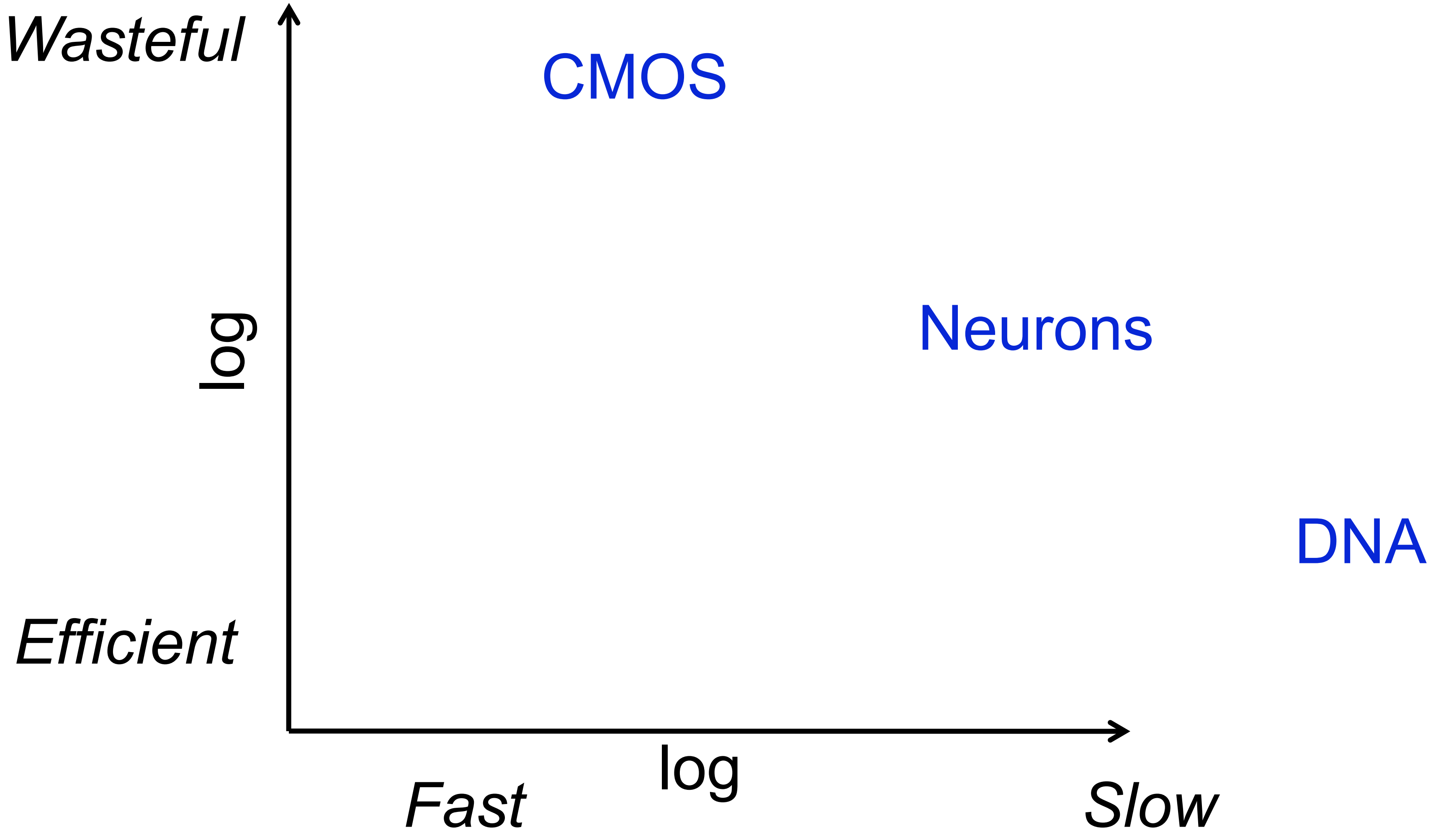



Wasteful
log
Efficient
Design tradeoffs
log
Fast
Slow

Tradeoffs are universal, but the details are not.

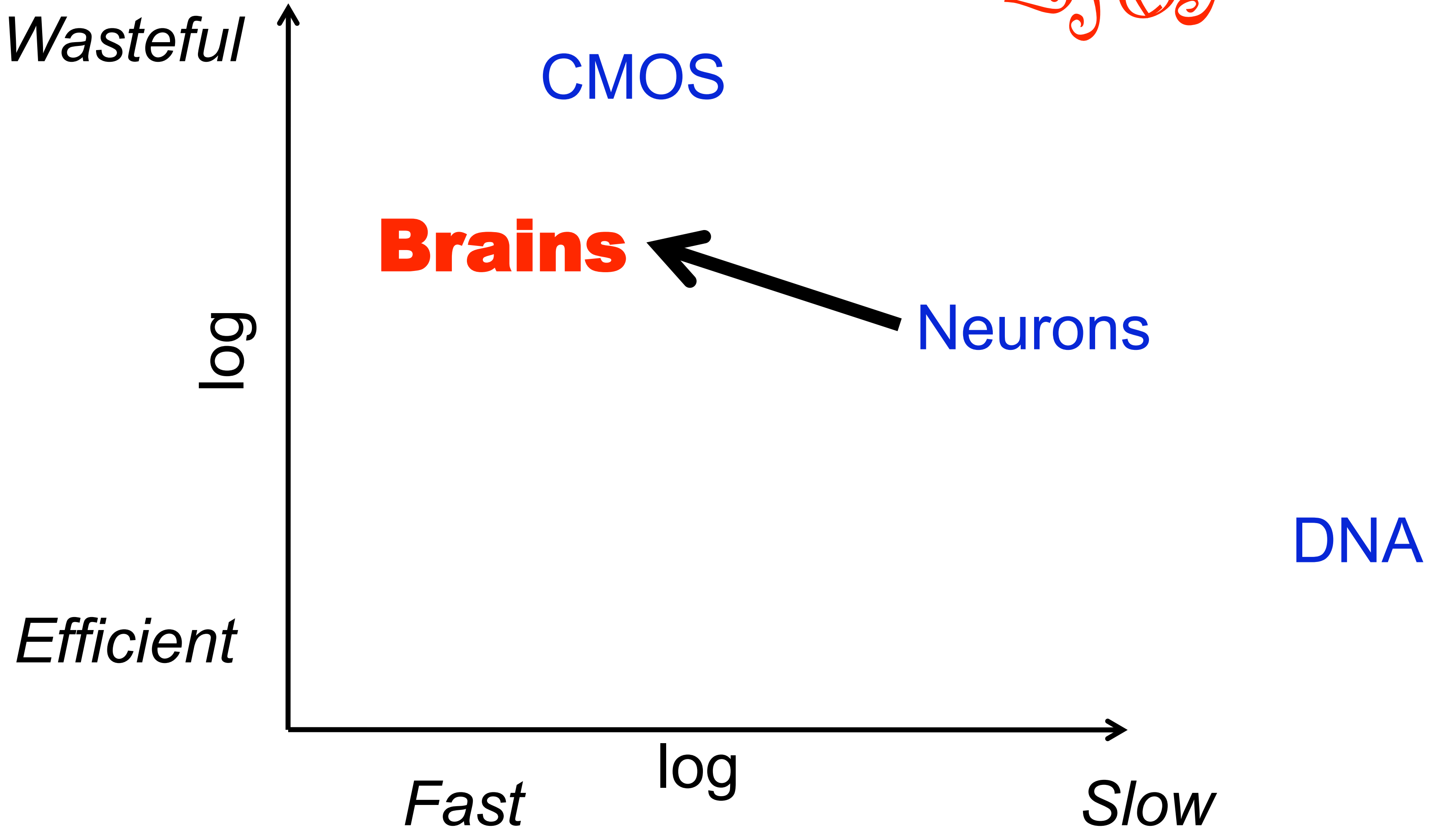


Computational hardware substrates



Some tasks: HARD for computers

EASY for us



Wasteful

CMOS

Brains

Neurons

DNA

log

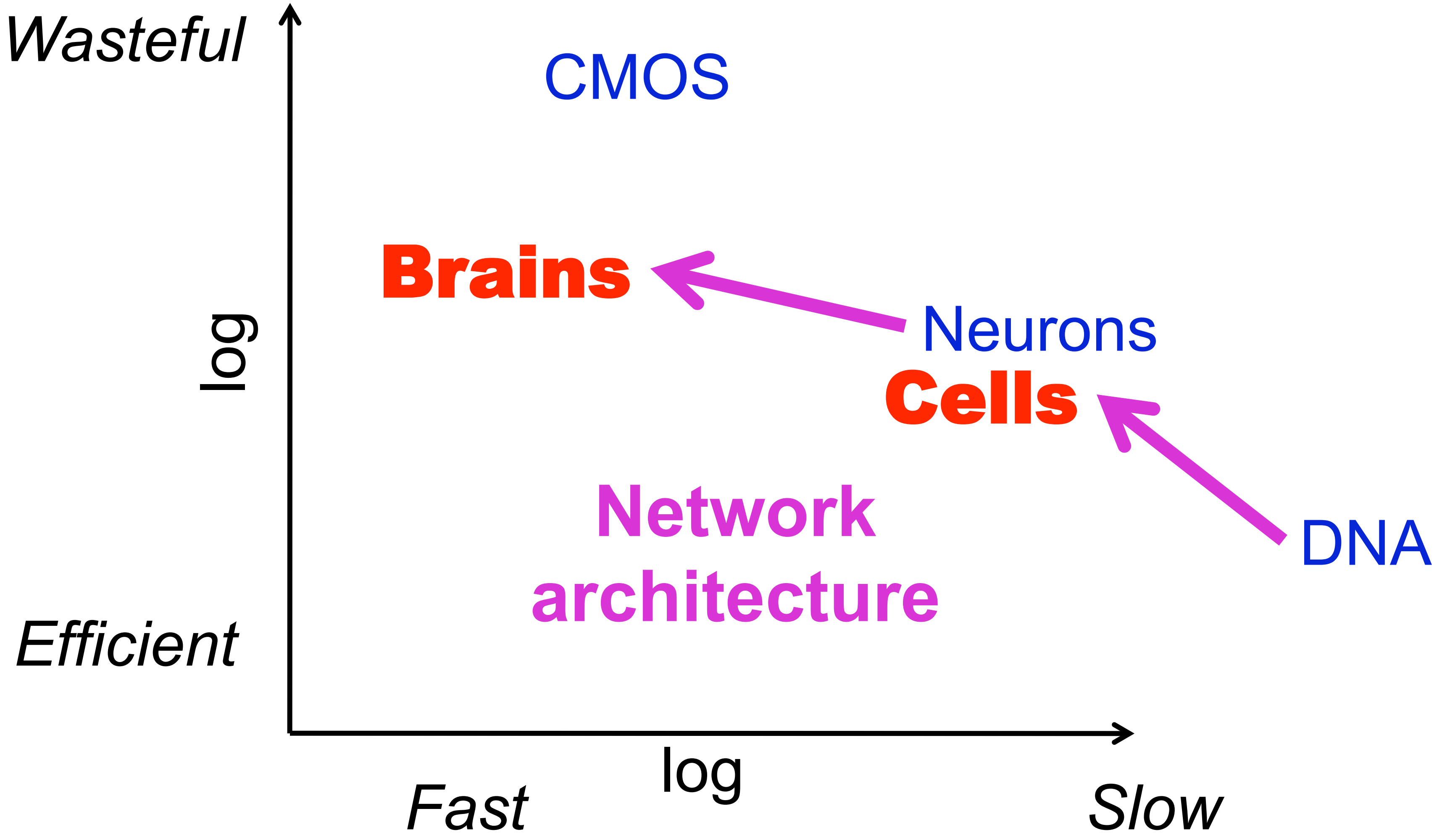
Efficient

log

Fast

Slow

What makes this possible?



Wasteful

CMOS

Brains

Neurons

Cells

DNA

log

Network architecture

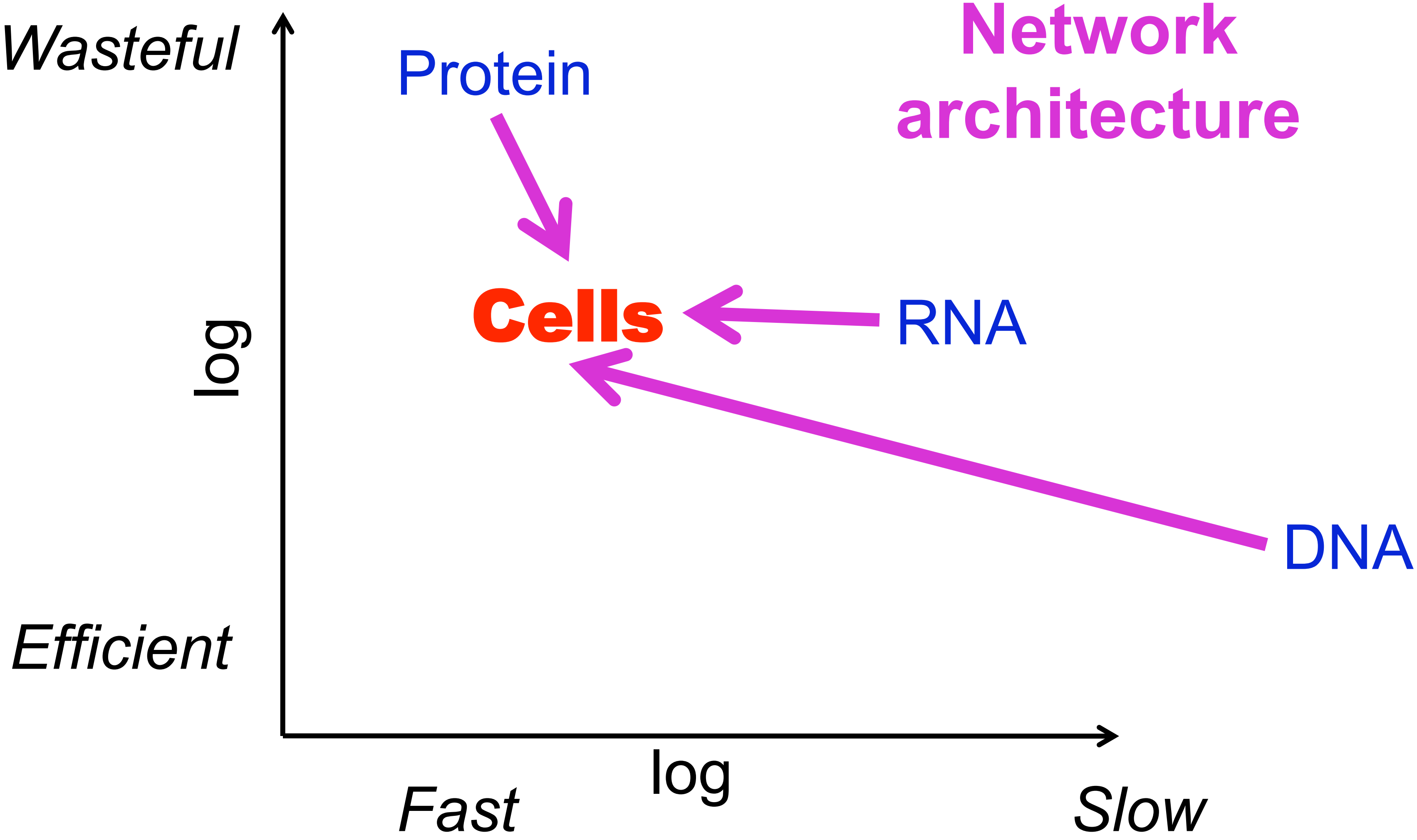
Efficient

log

Fast

Slow

Network architecture



Wasteful

Protein

Cells

RNA

log

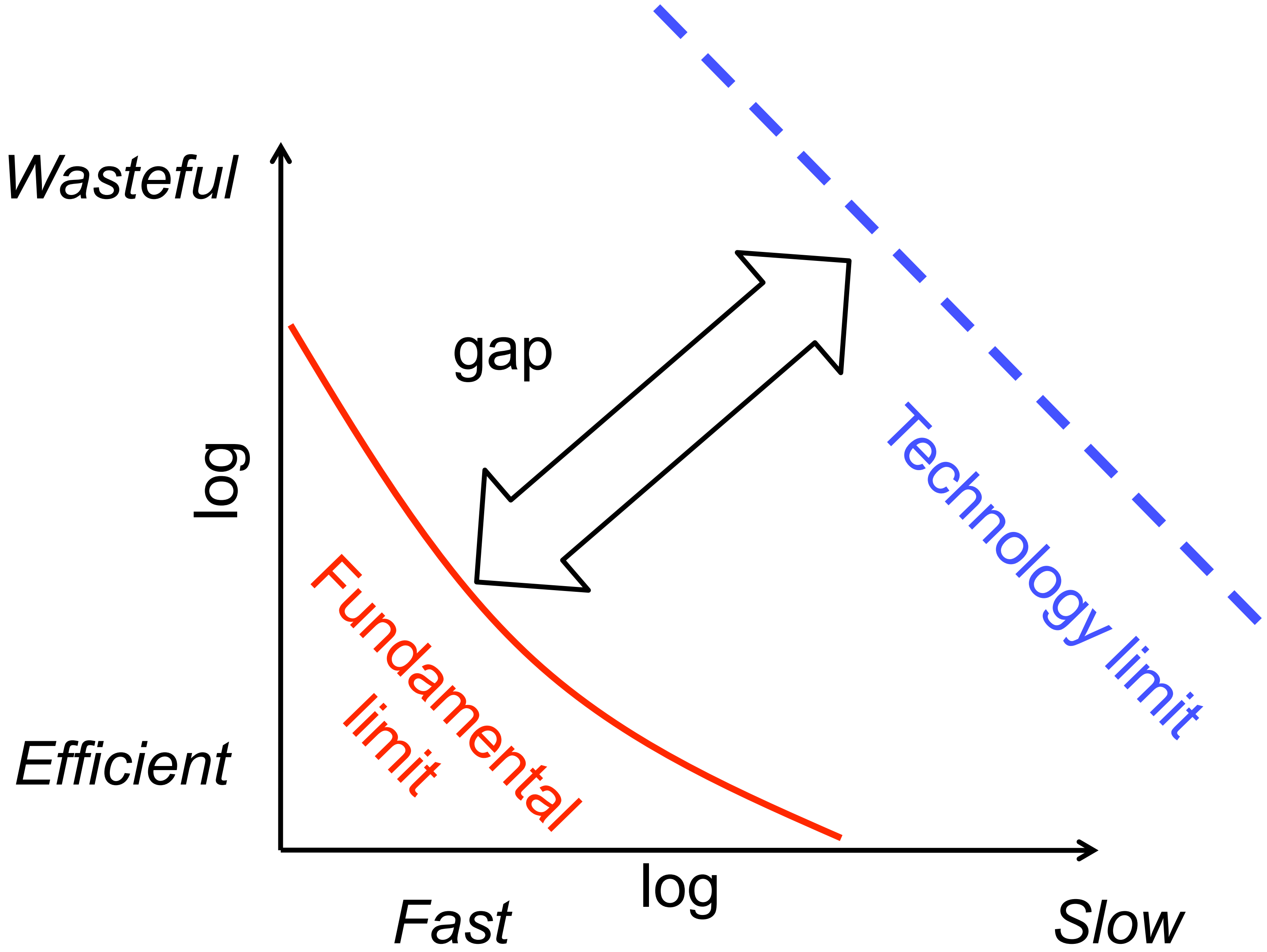
DNA

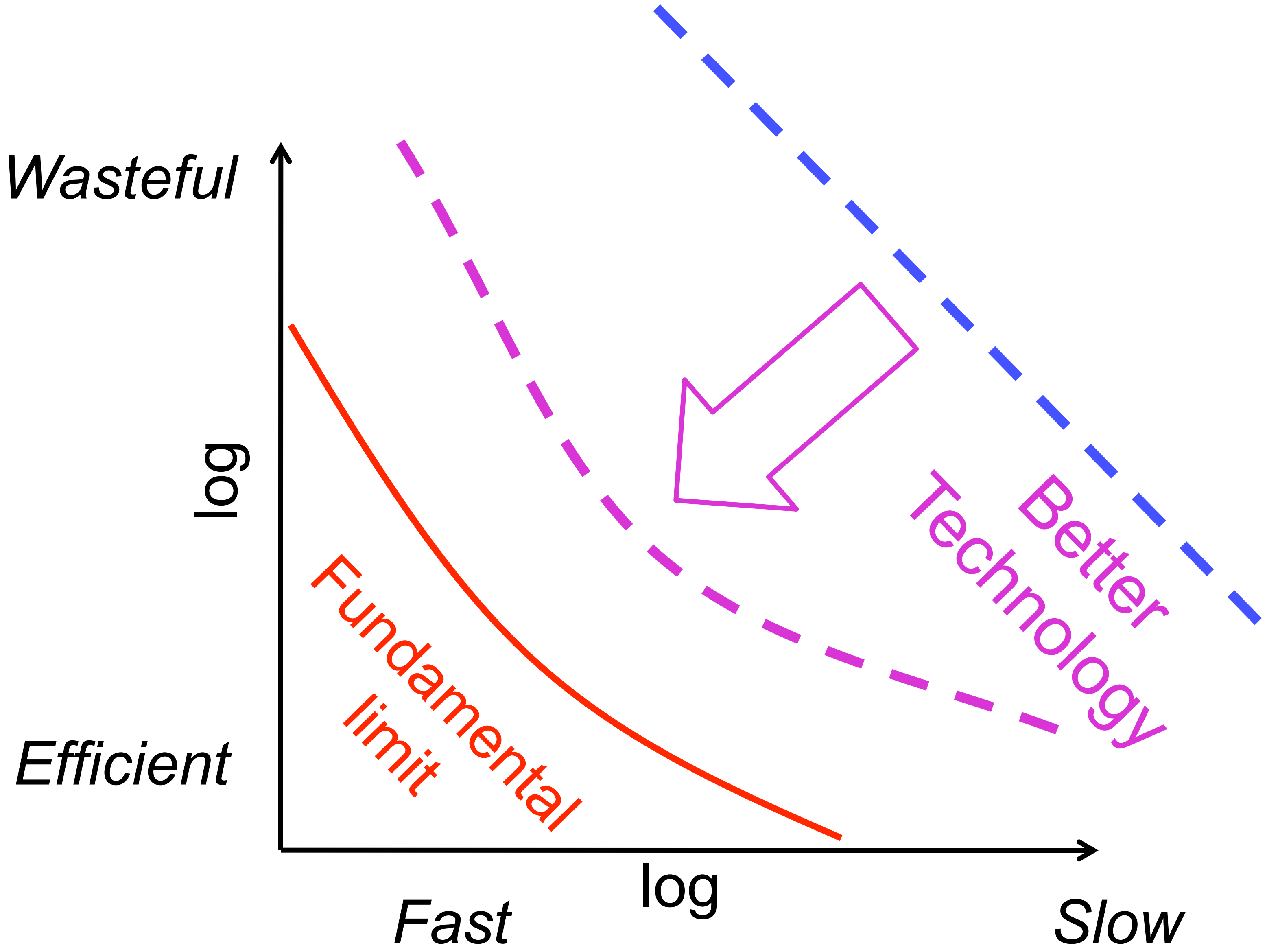
Efficient

log

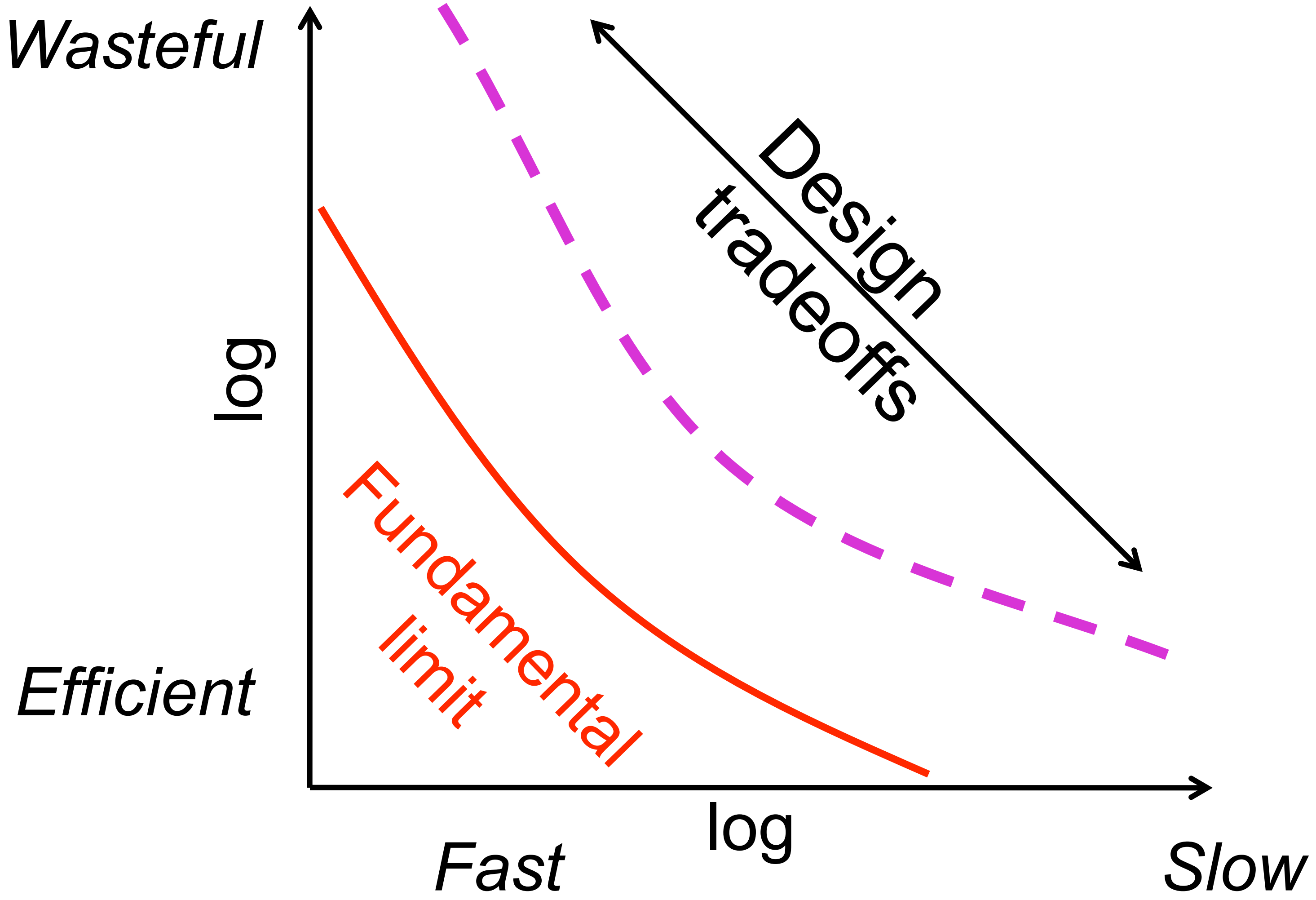
Fast

Slow


Wasteful
Efficient
log
gap
Technology limit
Fundamental limit
log
Fast
Slow


Wasteful
log
Efficient
Fundamental limit
Better Technology
log
Fast
Slow

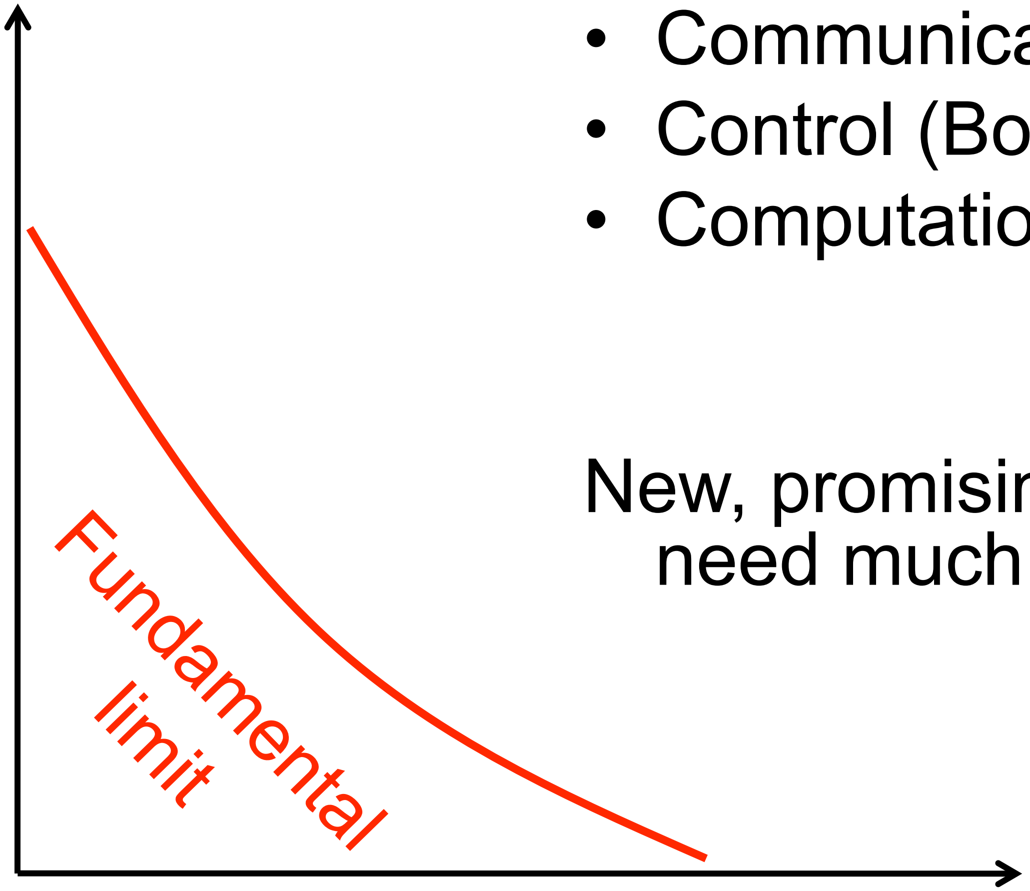
💣 bad ☹

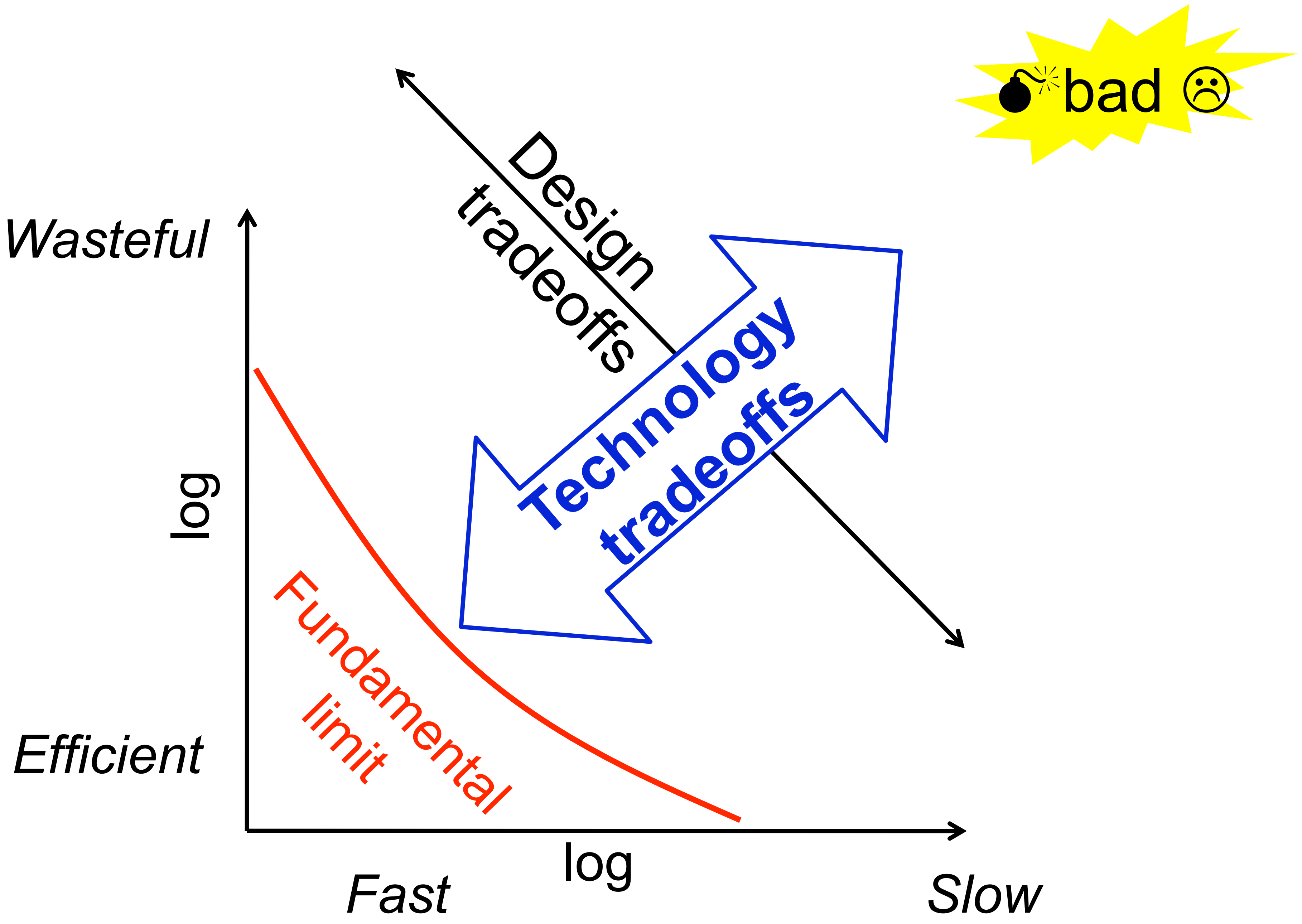

Wasteful
Efficient
log
Design tradeoffs
Fundamental limit
log
Fast
Slow

Existing hard limits have restrictive assumptions and few dimensions

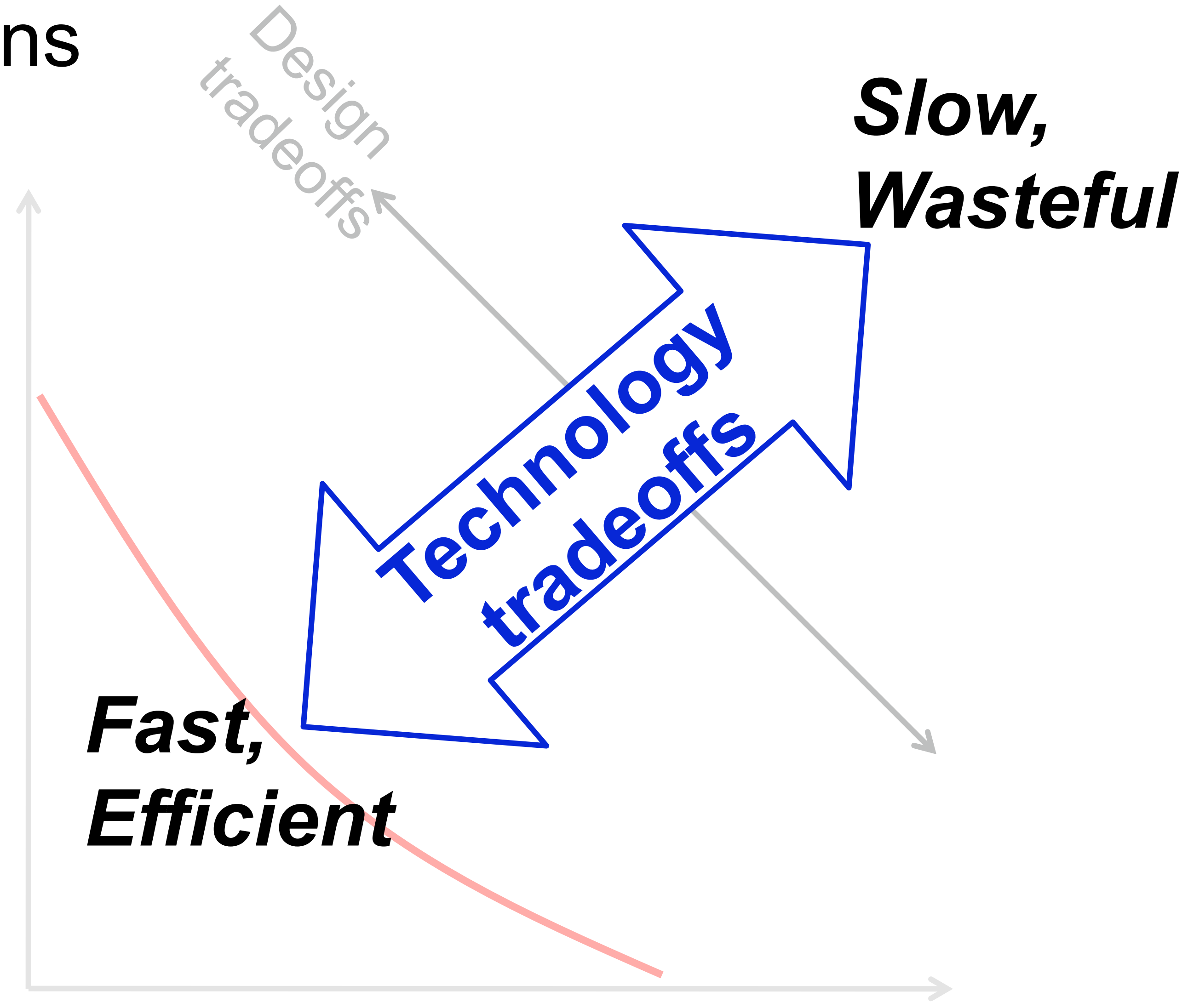
- Thermodynamics (Carnot)
- Communications (Shannon)
- Control (Bode)
- Computation (Turing)

New, promising unifications but need much more




💣bad ☹
Design tradeoffs
Technology tradeoffs
Wasteful
log
Efficient
Fundamental limit
Fast
log
Slow

Collapse dimensions



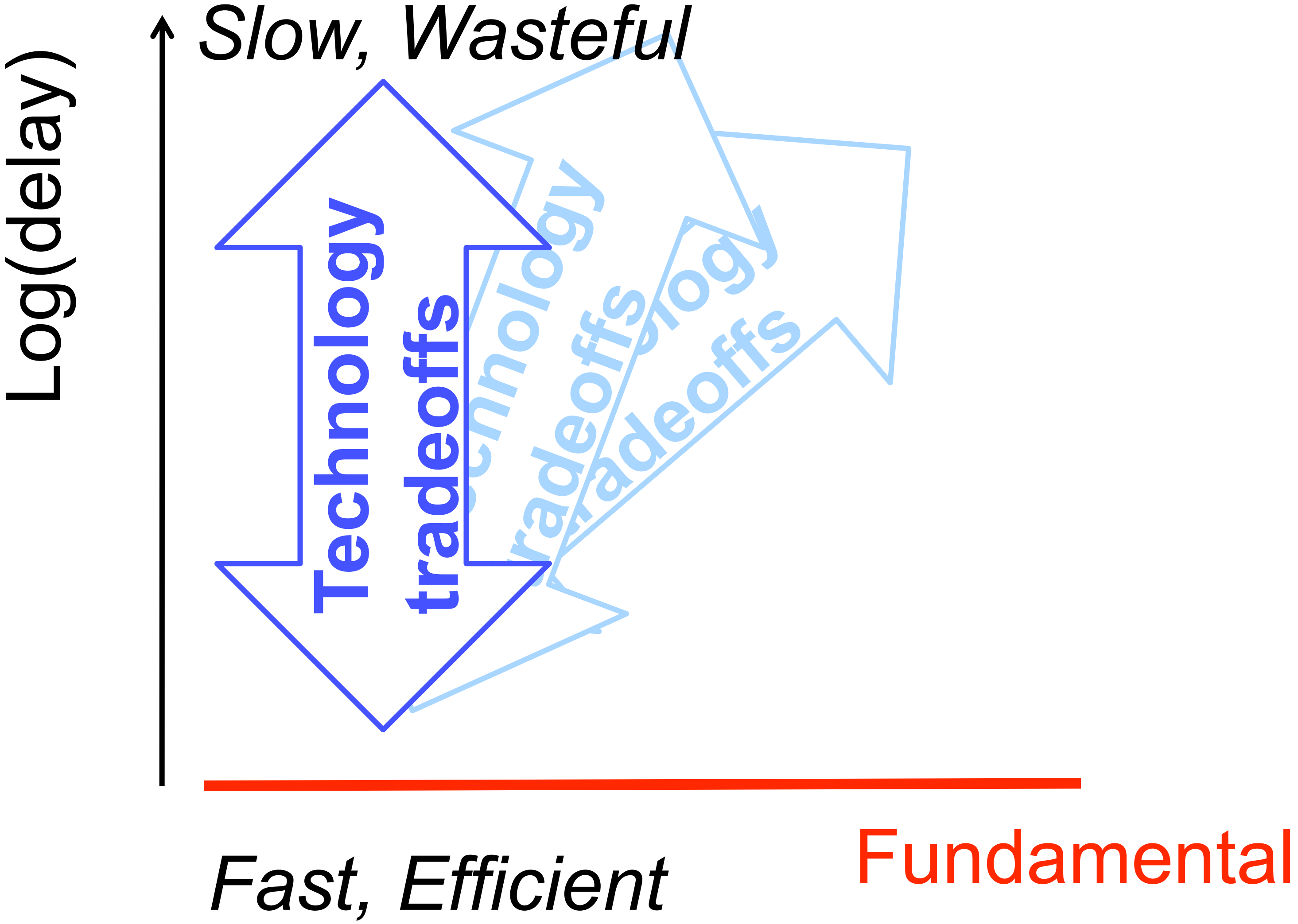
Design tradeoffs

Slow, Wasteful

Technology tradeoffs

Fast, Efficient

Collapse dimensions



Log(delay)

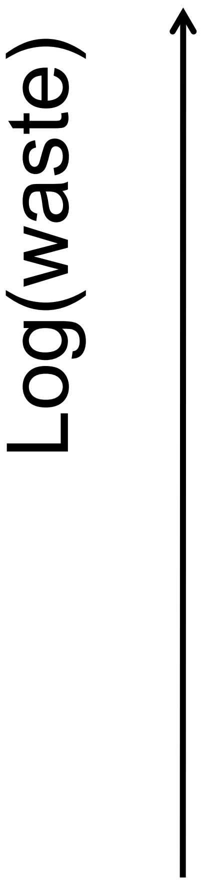
Slow, Wasteful

Technology tradeoffs

Fast, Efficient

Fundamental

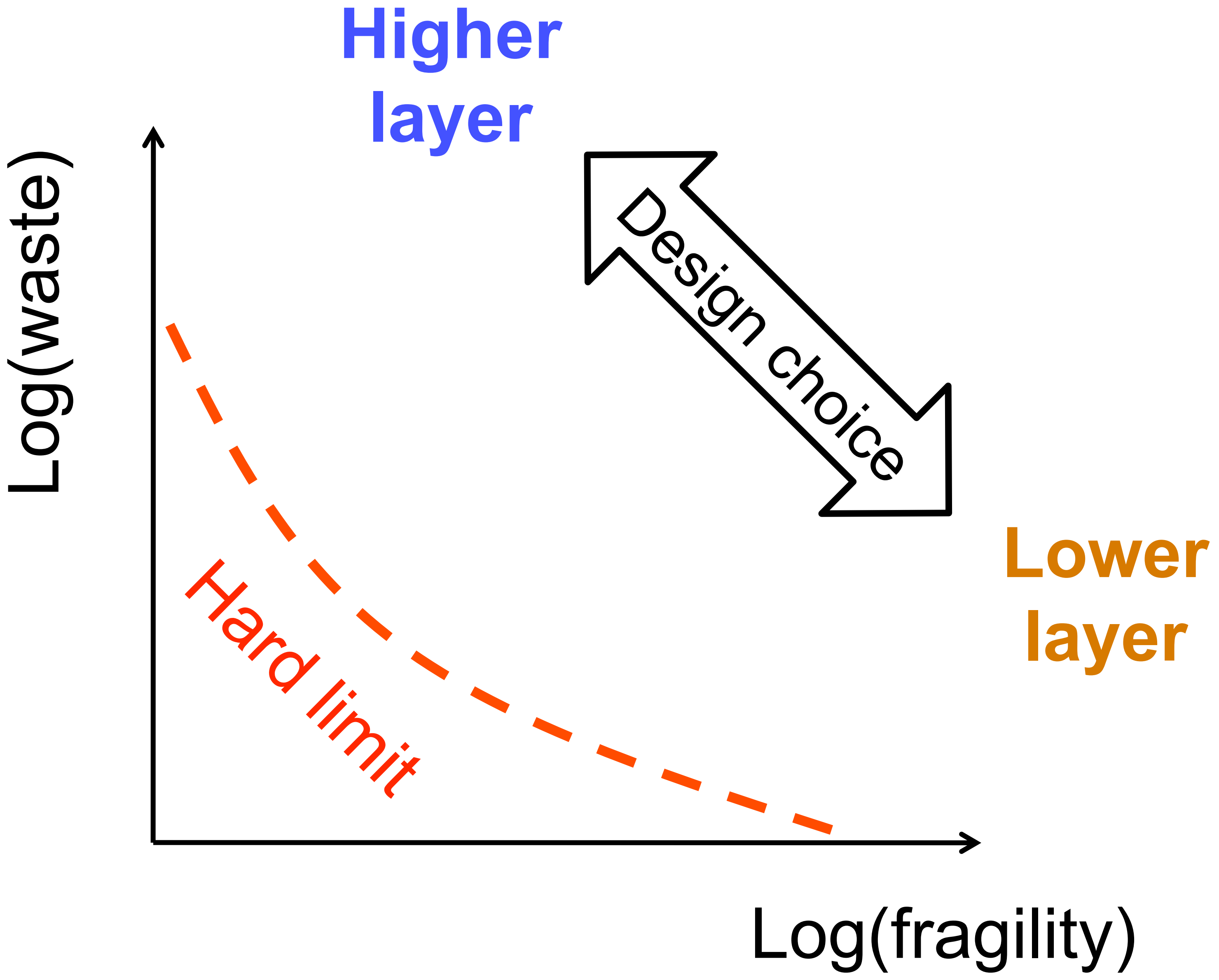
Slow, Wasteful



Fast, Efficient

Waste

- time
- energy
- materials
- …


Higher layer
Design choice
Lower layer
Log(waste)
Hard limit
Log(fragility)

[**a system**]
can be *robust*
for a given
[**property**]
and a set of
[**perturbations**]

Yet
be ***fragile for***
a different
[**property**]
or
[**perturbation**]

Log(robustness) → Log(fragility)

Question: Human complexity

Robust	Yet Fragile
☺ Efficient, flexible metabolism	☹ Obesity and diabetes
☺ Regeneration & renewal	☹ Cancer
☺ Rich microbial symbionts	☹ Parasites, infection
☺ Immune systems	☹ Inflammation, Auto-Im.
🗐 Complex societies	☠ Epidemics, war, …
⊞ Advanced technologies	💣 Catastrophic failures

Mechanism?

Robust

- ☺ Efficient, flexible metabolism
- ☺ Regeneration & renewal
  - ☺ Fat accumulation
  - ☺ Insulin resistance
  - ☺ Inflammation

Fluctuating energy

Yet Fragile

- ☹ Obesity and diabetes
- ☹ Cancer
  - ☹ Fat accumulation
  - ☹ Insulin resistance
  - ☹ Inflammation

Static energy

Implications/ Generalizations

	Robust		Yet Fragile
☺	Efficient, flexible metabolism	☹	Obesity and diabetes
☺	Rich microbial symbionts	☹	Parasites, infection
☺	Immune systems	☹	Inflammation, Auto-Im.
☺	Regeneration & renewal	☹	Cancer
🗐	Complex societies	☠	Epidemics, war, …
⊞	Advanced technologies	💣	Catastrophic failures

- Fragility = Hijacking, side effects, unintended… of mechanisms evolved for robustness
- Complexity is driven by control, robust/fragile tradeoffs
- Math: New robust/fragile conservation laws

- Resilience/safety/security Engineering/Economics: “Human error” and “human nature” is often a symptom of bad system architecture

Other dimensions

Robust

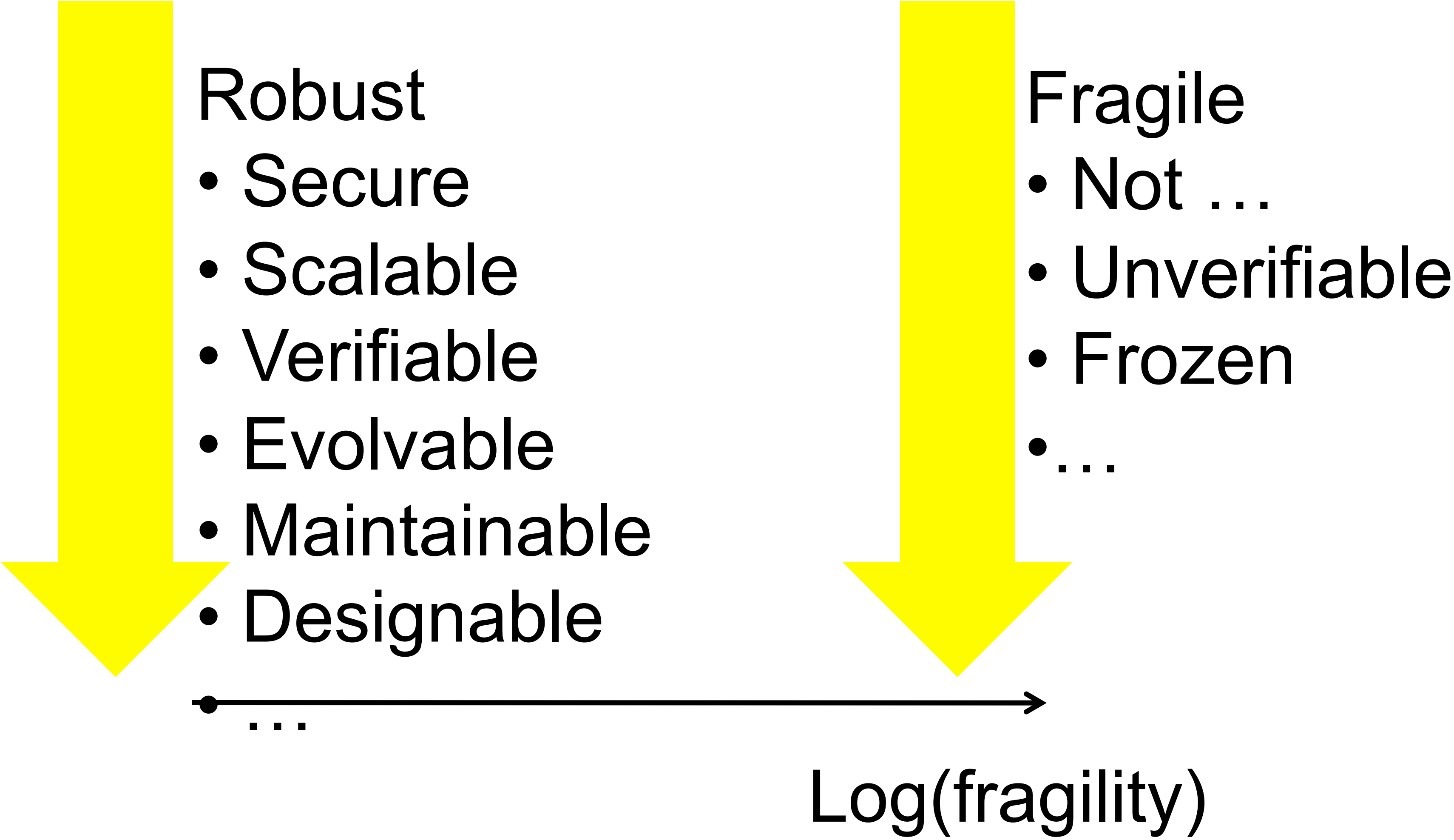
- Secure
- Scalable
- Verifiable
- Evolvable
- Maintainable
- Designable
- …

Fragile

- Insecure
- Not scalable
- Unverifiable
- Frozen
- …

Log(fragility)

Collapse other dimensions



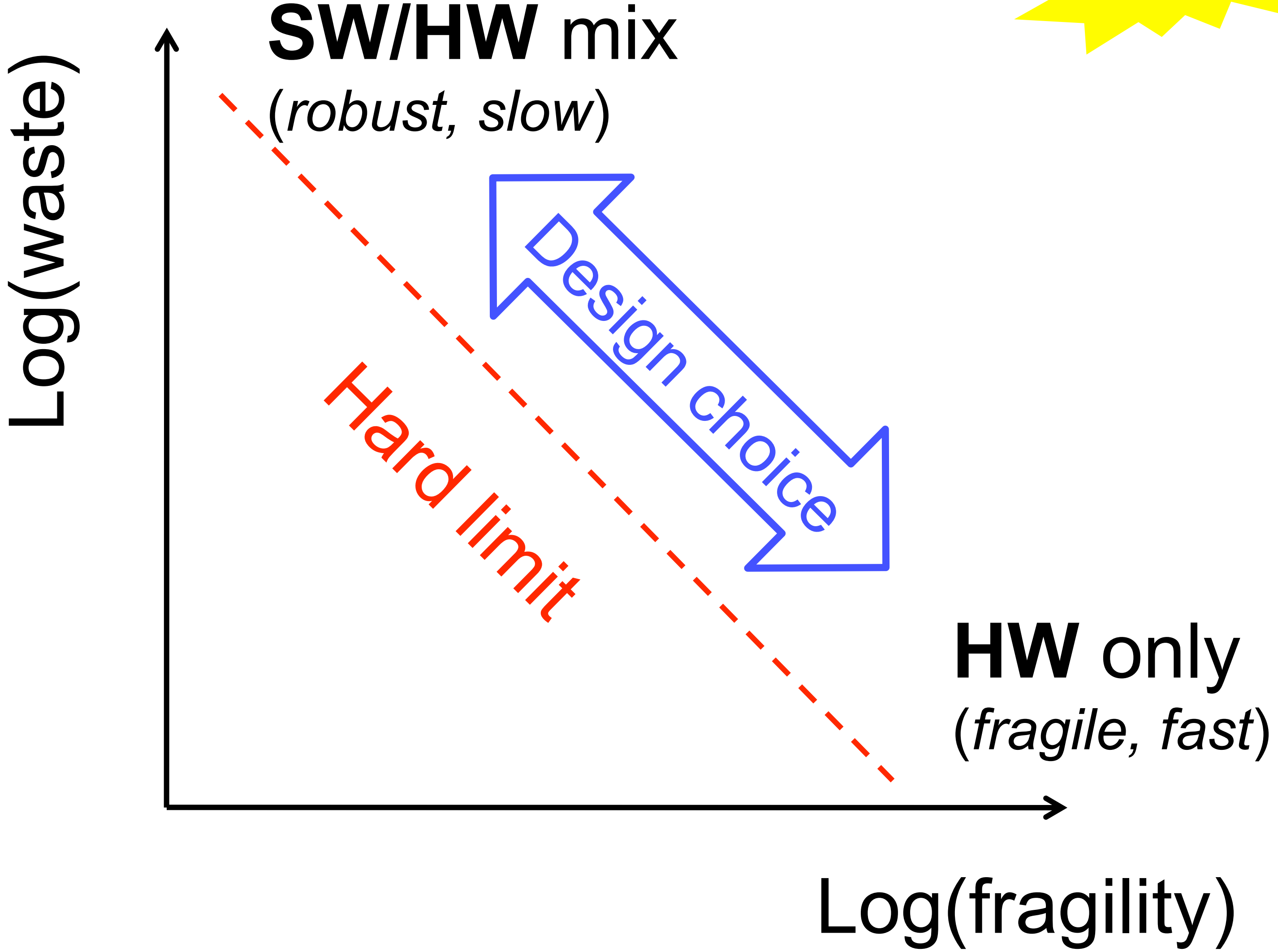
Robust
- Secure
- Scalable
- Verifiable
- Evolvable
- Maintainable
- Designable
- …

Fragile
- Not …
- Unverifiable
- Frozen
- …

Log(fragility)

Log(fragility)

💣bad ☹



SW/HW mix

(*robust, slow*)

Design choice

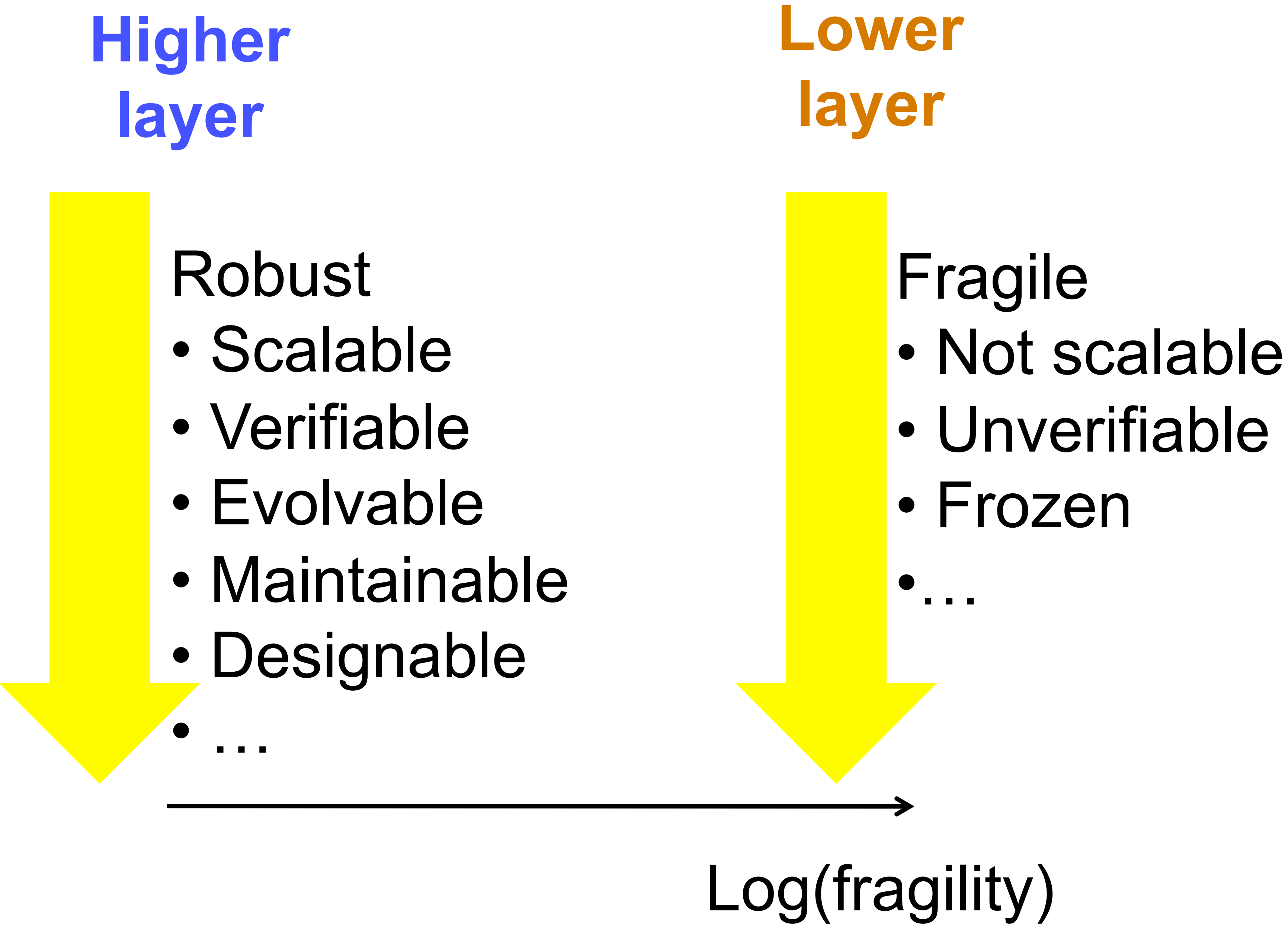
Hard limit

HW only

(*fragile, fast*)

Log(waste)

Log(fragility)



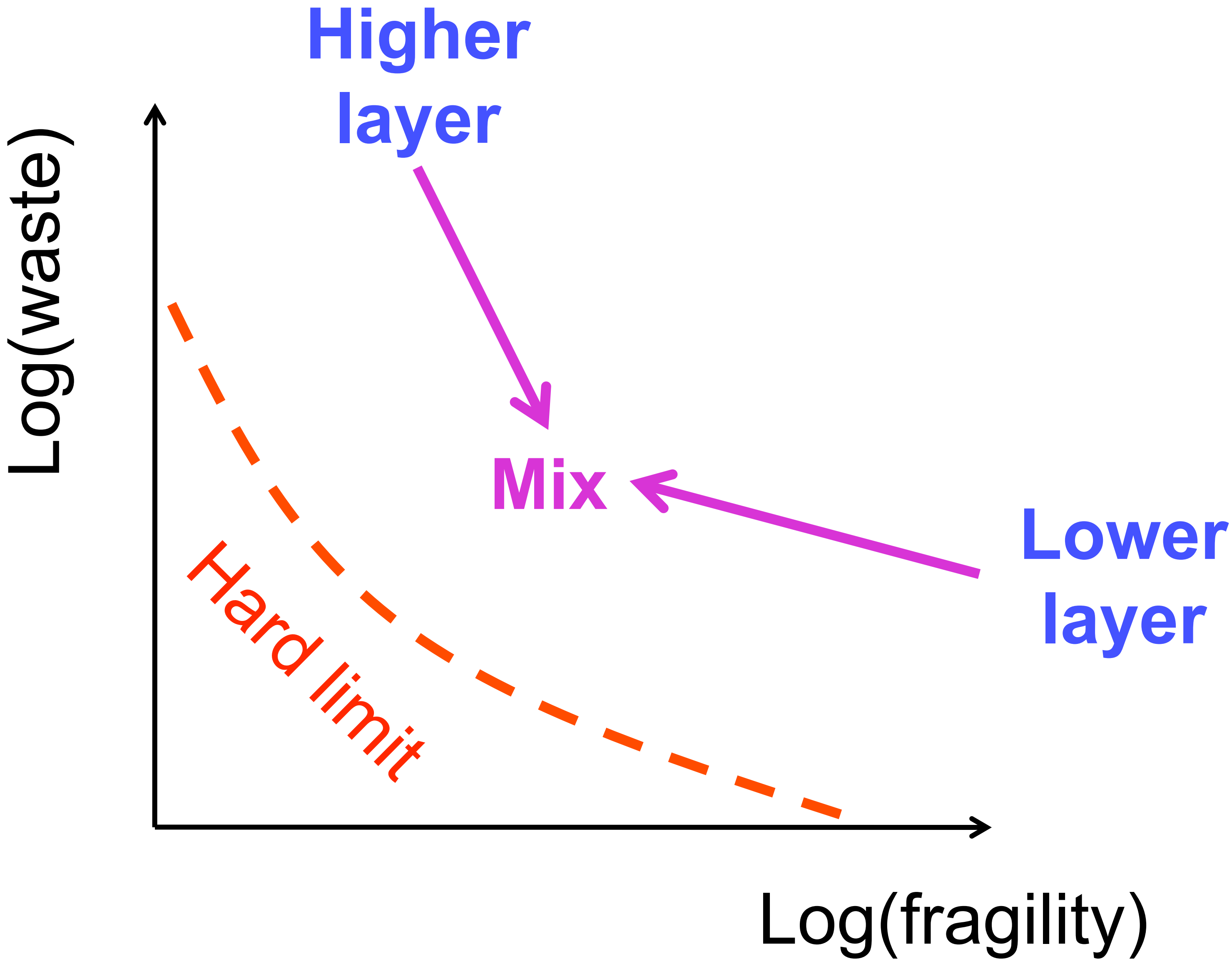
Higher layer

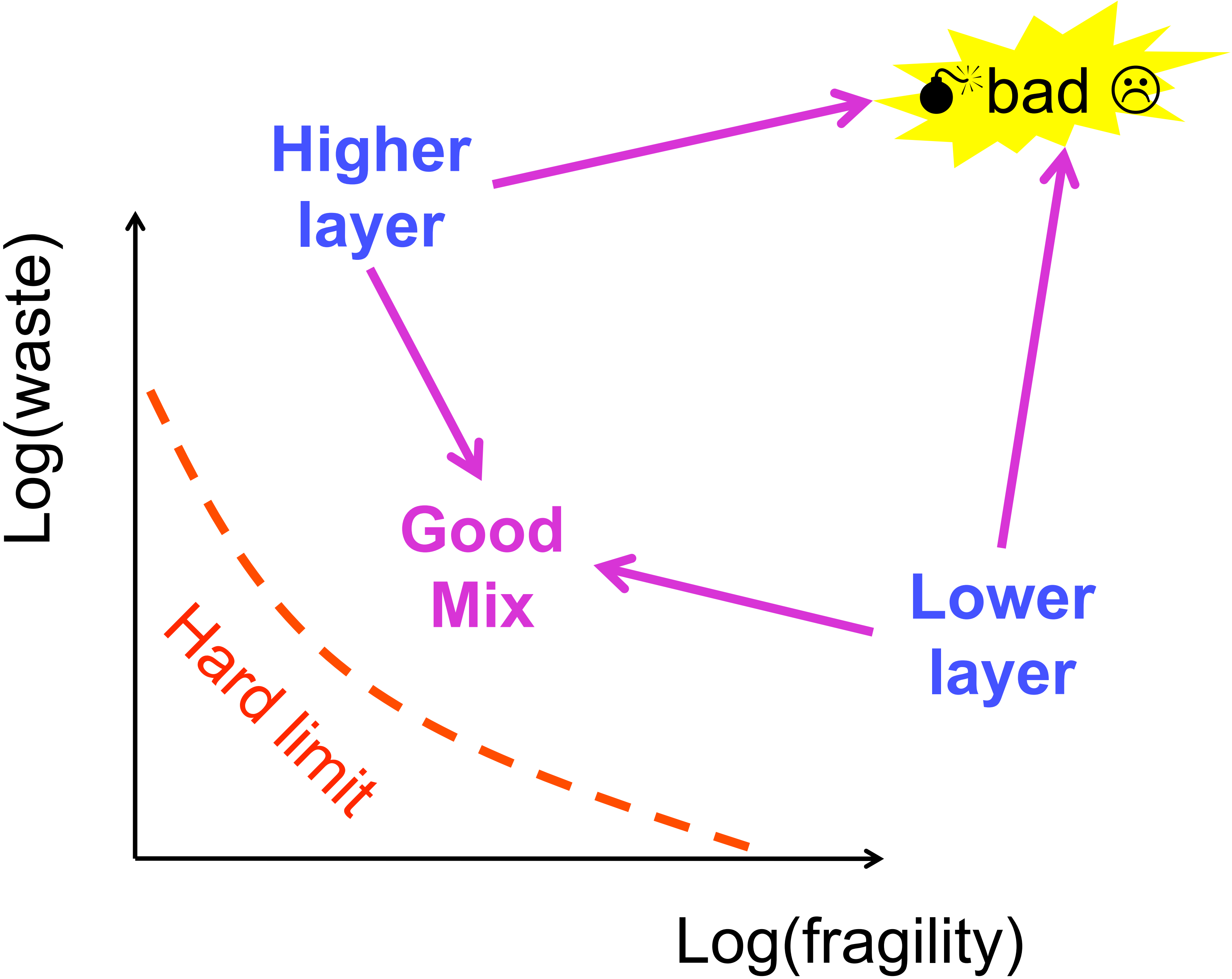
Robust
- Scalable
- Verifiable
- Evolvable
- Maintainable
- Designable
- …

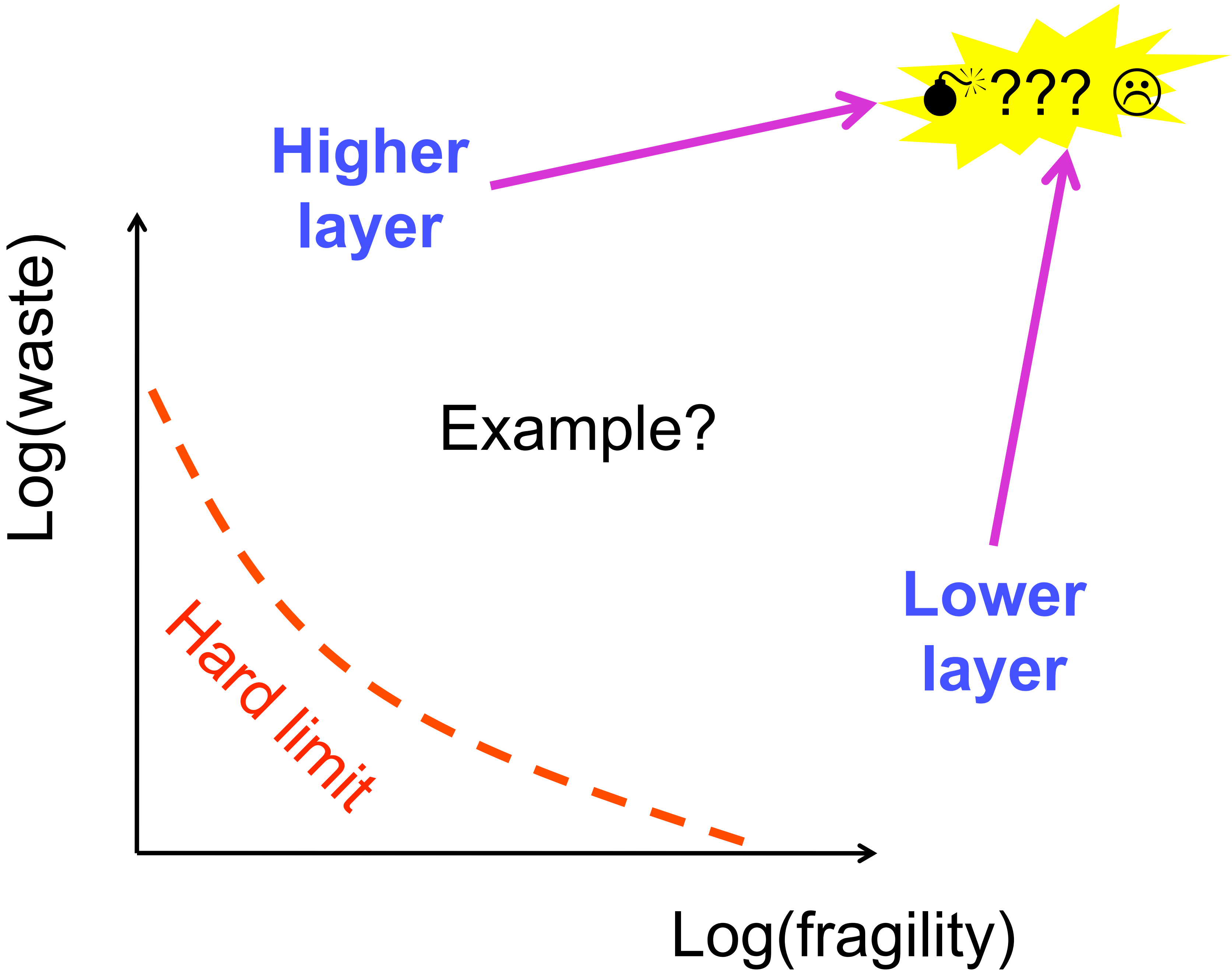
Lower layer

Fragile
- Not scalable
- Unverifiable
- Frozen
- …

Log(fragility)


Higher layer
Mix
Lower layer
Log(waste)
Hard limit
Log(fragility)


💣bad ☹
Higher layer
Good Mix
Lower layer
Hard limit
Log(waste)
Log(fragility)



💣??? ☹

Higher layer

Example?

Lower layer

Log(waste)

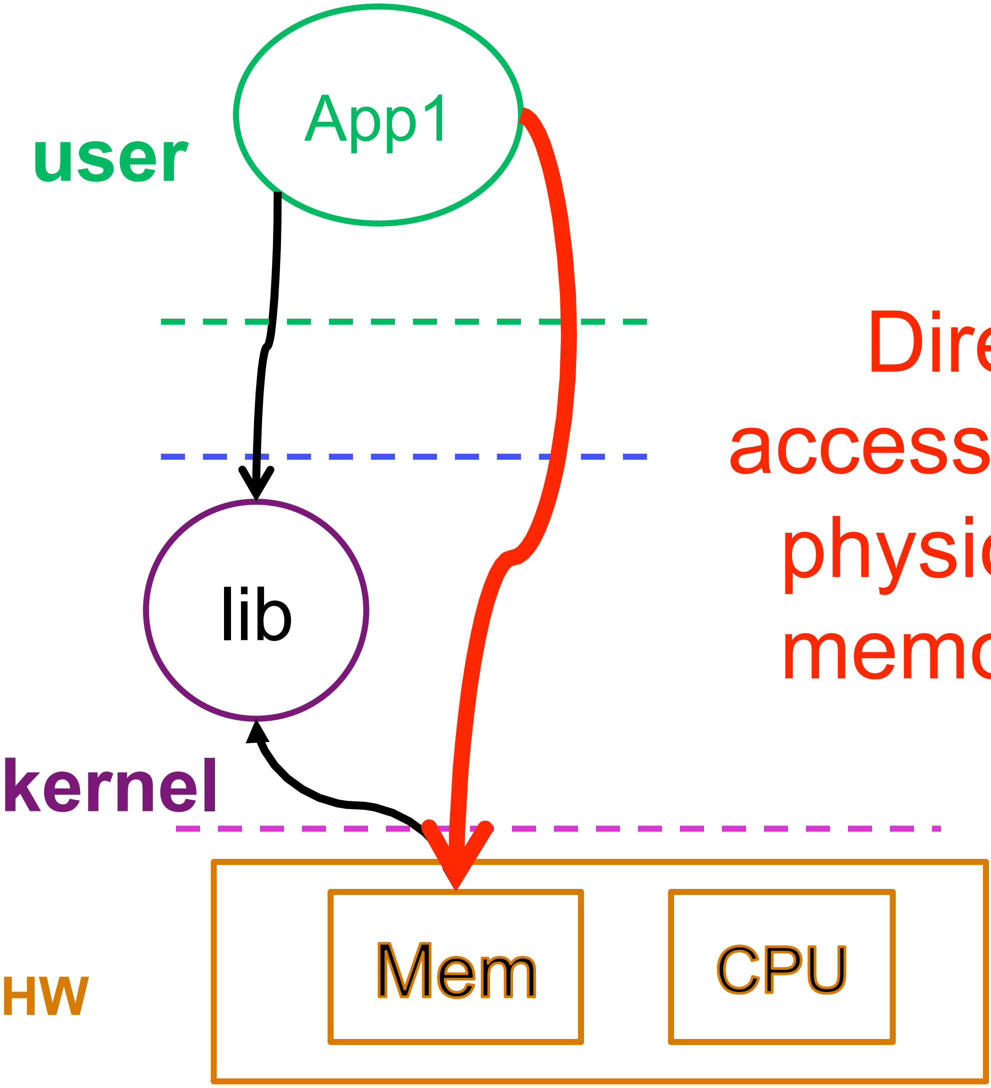
Hard limit

Log(fragility)

Don’t cross layers

💣??? ☹

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

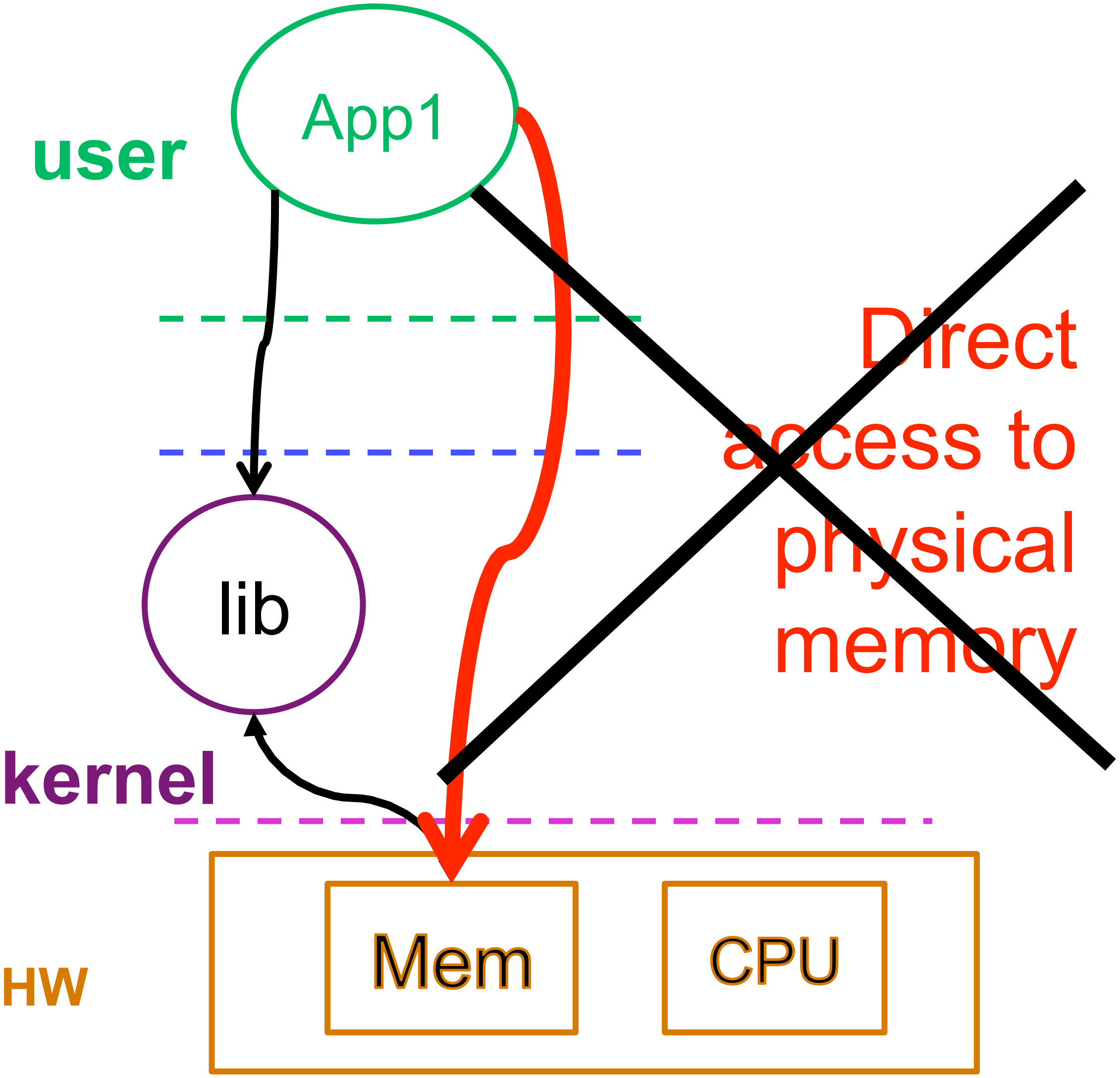


Direct access to physical memory

Separate logical names and physical addresses

💣 ??? ☹

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



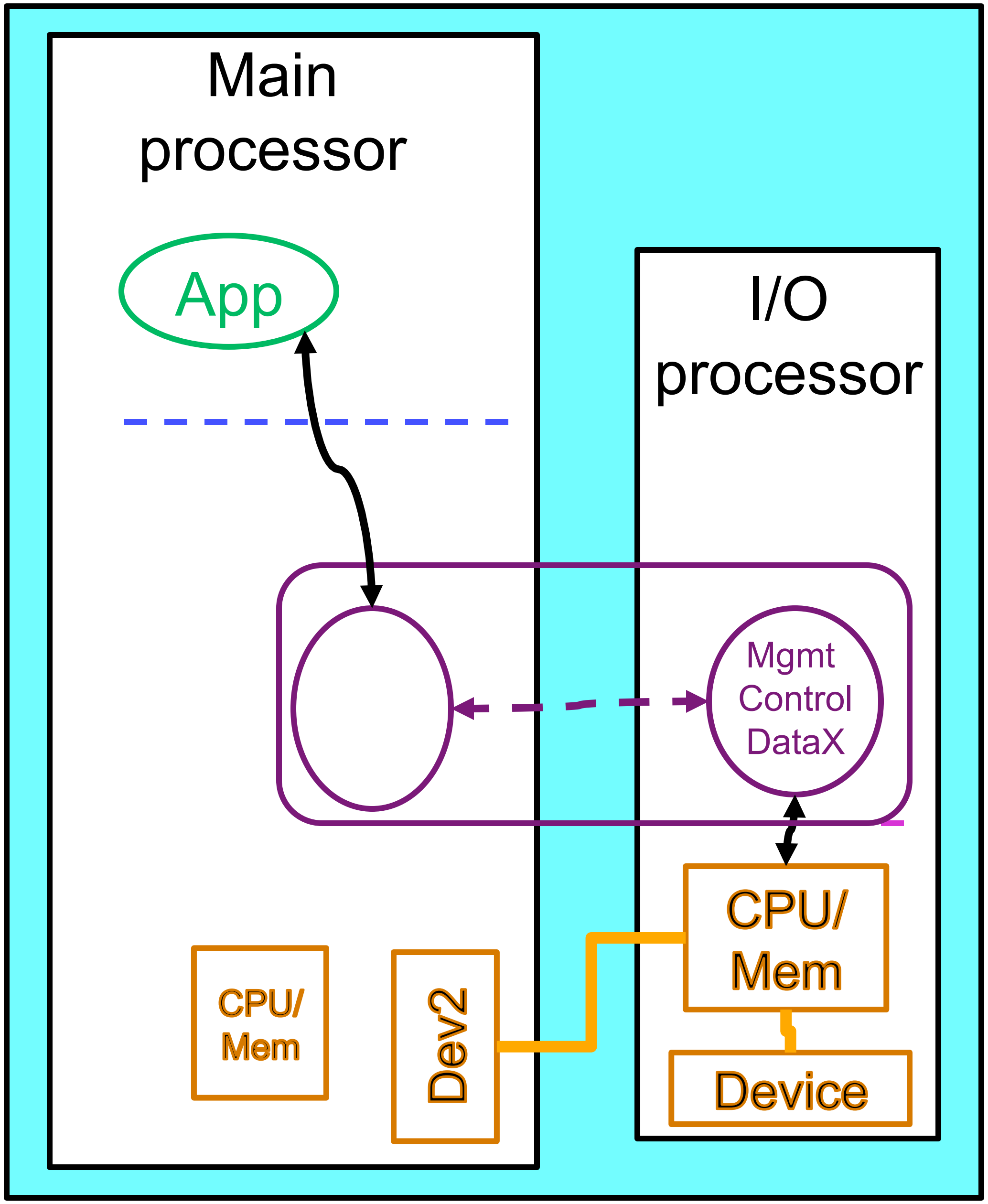
Separate logical names and physical addresses

Naming and addressing are important topics in OS

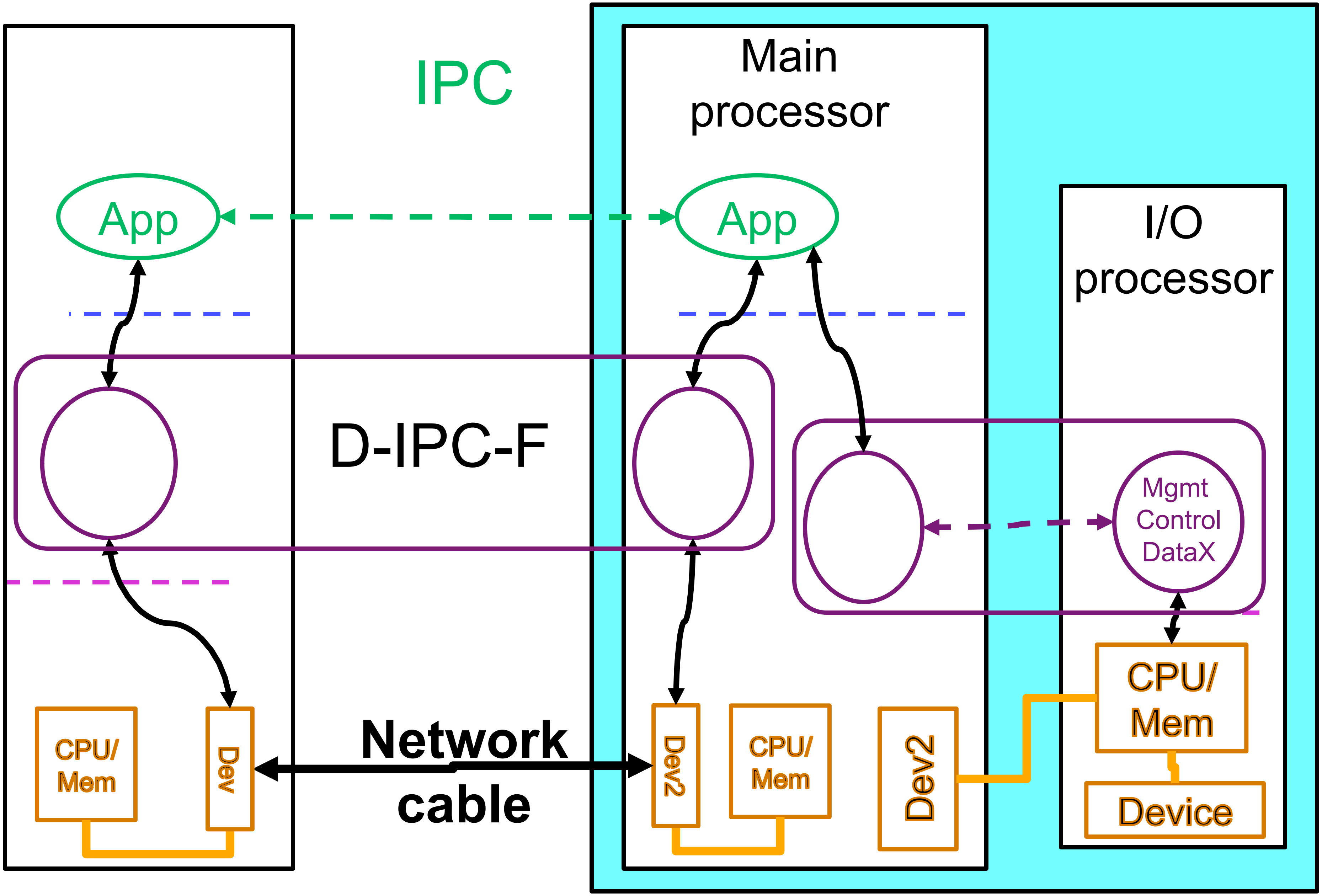
Needs to be an even richer topic in networking

So, finally, let’s look at a minimal network

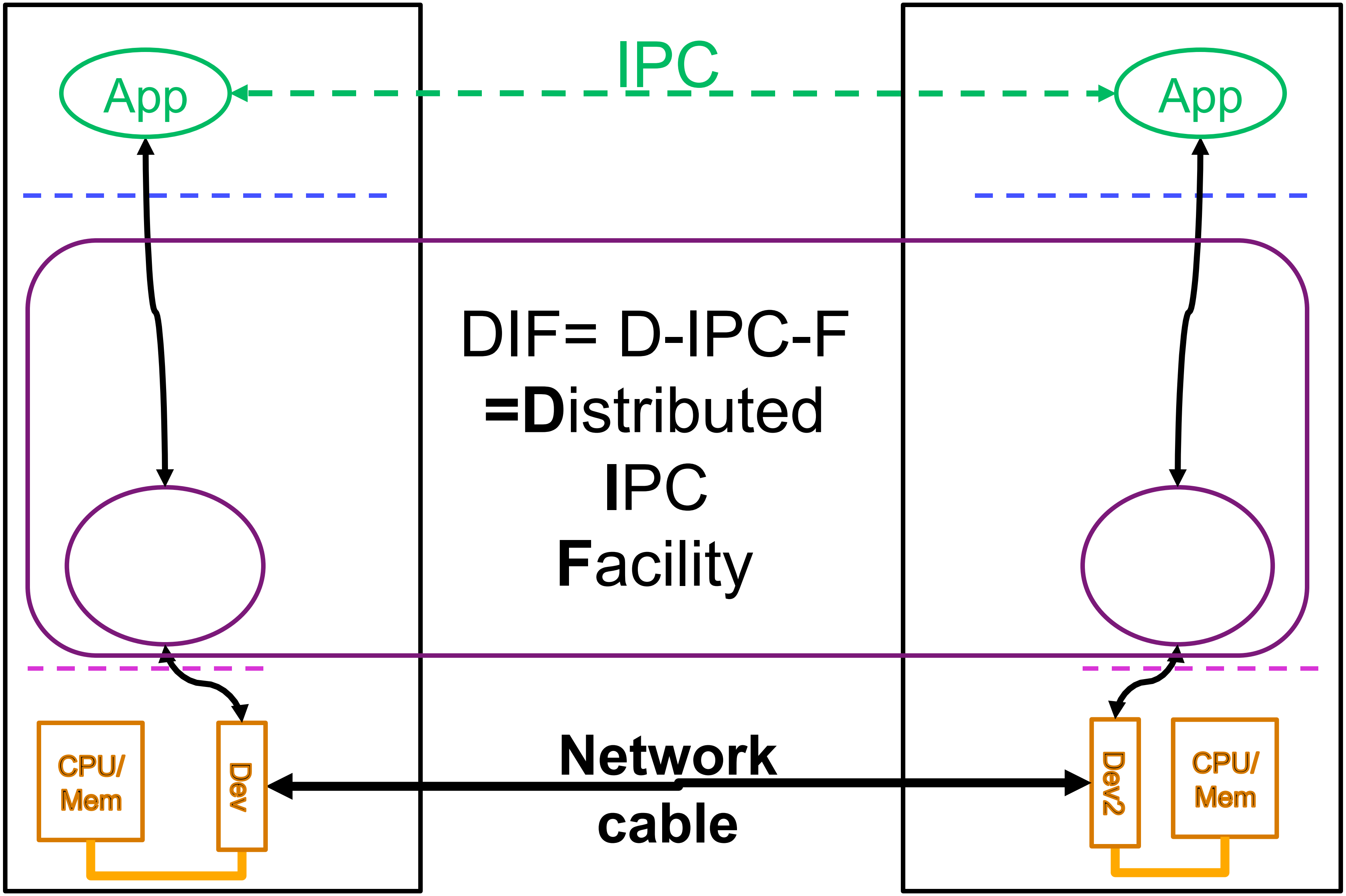
Might be all in the same “box”.



A network with another “box”…



A minimal network without a NIC.



DIF= D-IPC-F

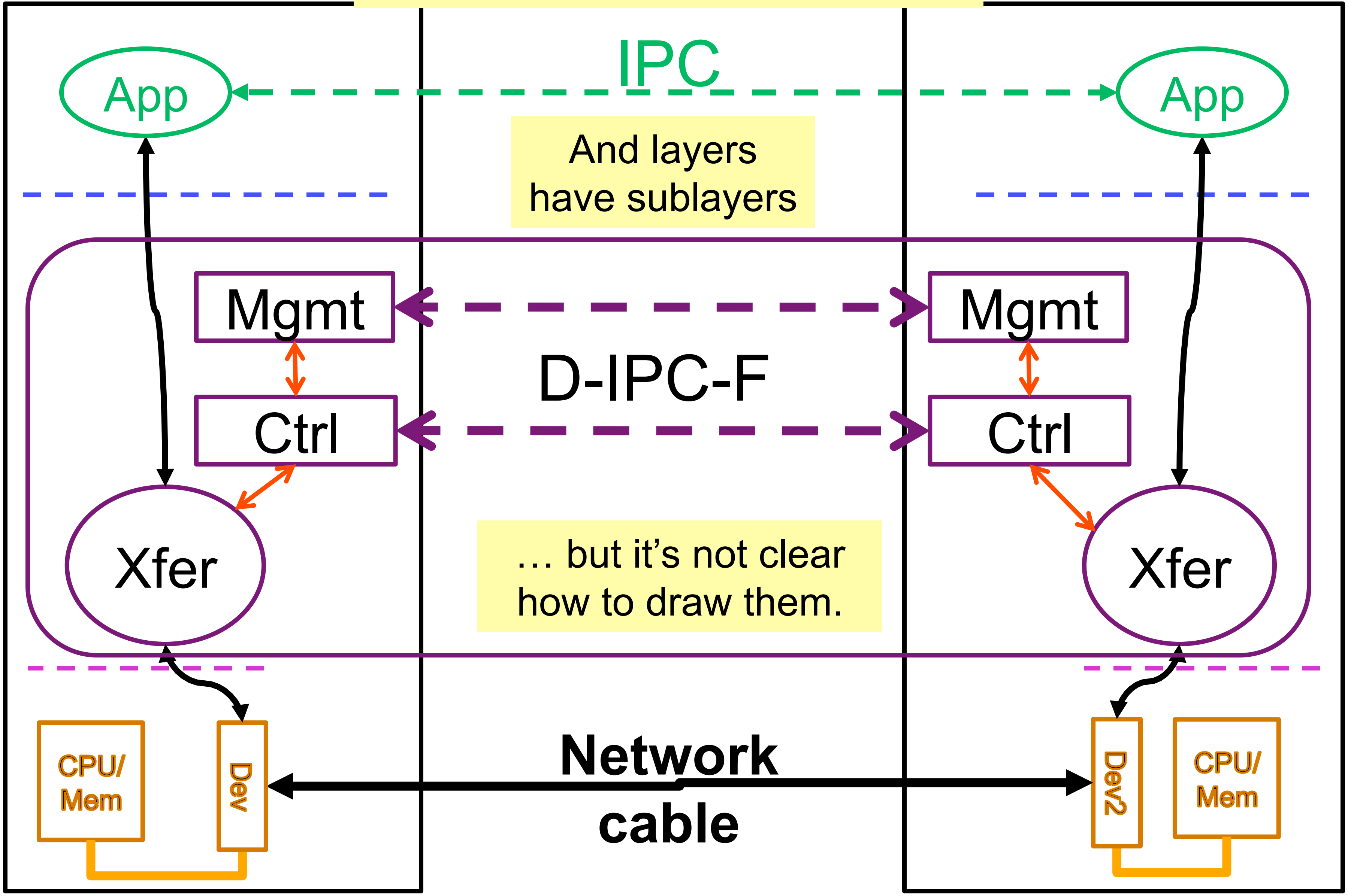
=**D**istributed

IPC

Facility

Network cable

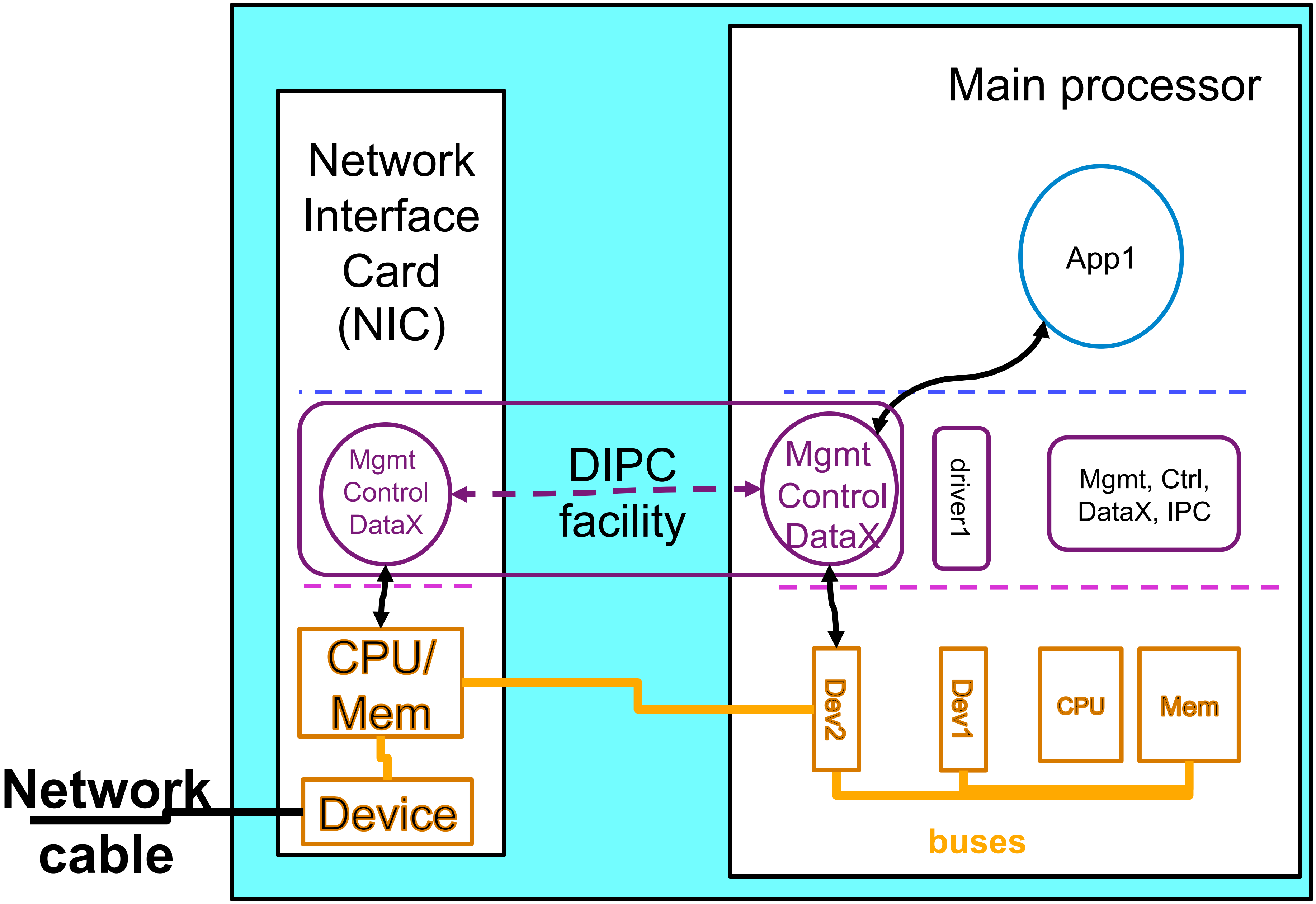
Mgmt and Cntrl become even more complex



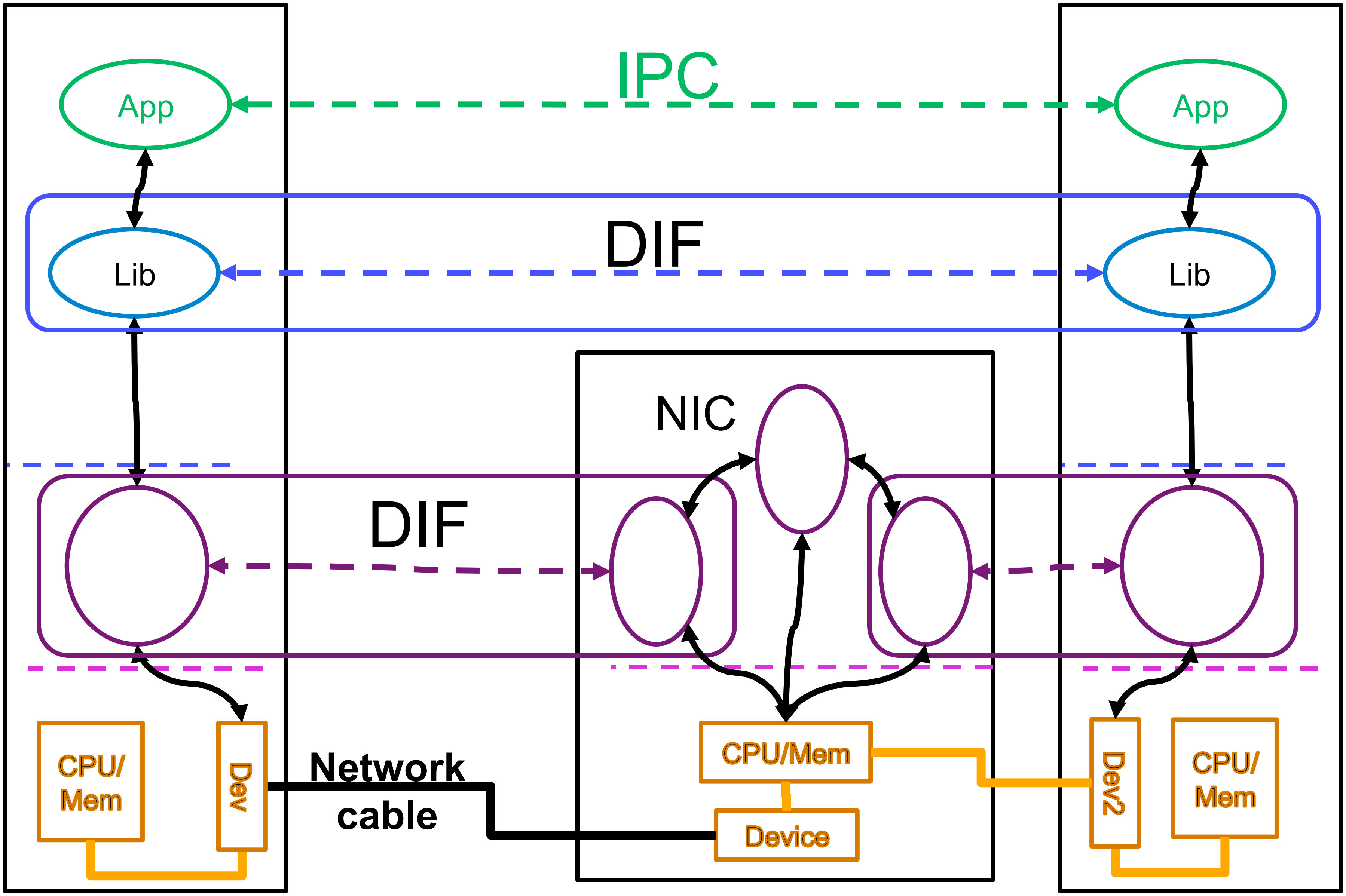
And layers have sublayers

… but it's not clear how to draw them.

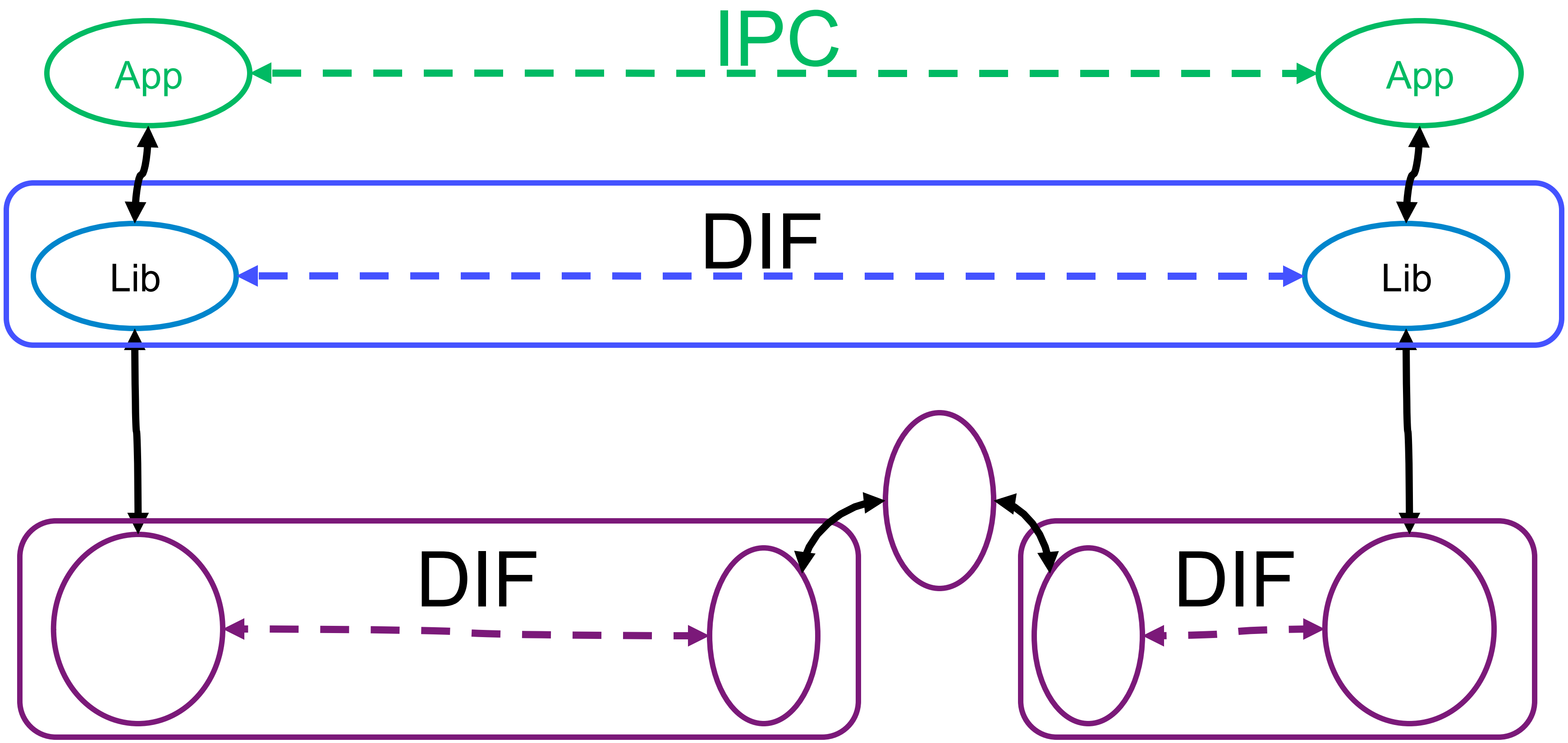
What is a NIC?


Main processor
Network
Interface
Card
(NIC)
App1
Mgmt
Control
DataX
DIPC
facility
Mgmt
Control
DataX
driver1
Mgmt, Ctrl,
DataX, IPC
CPU/
Mem
Dev2
Dev1
CPU
Mem
Device
Network
cable
buses

A minimal network with a NIC

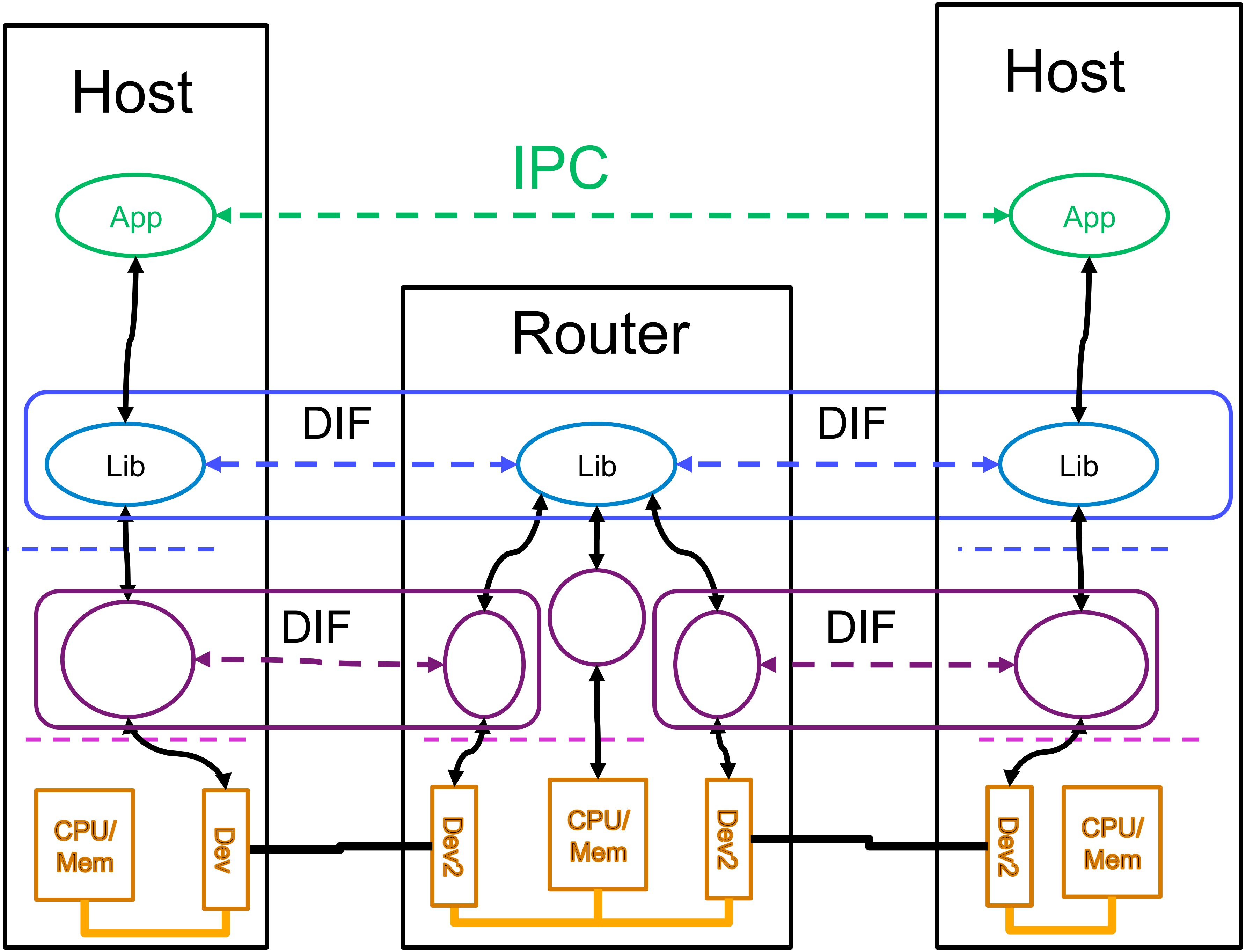


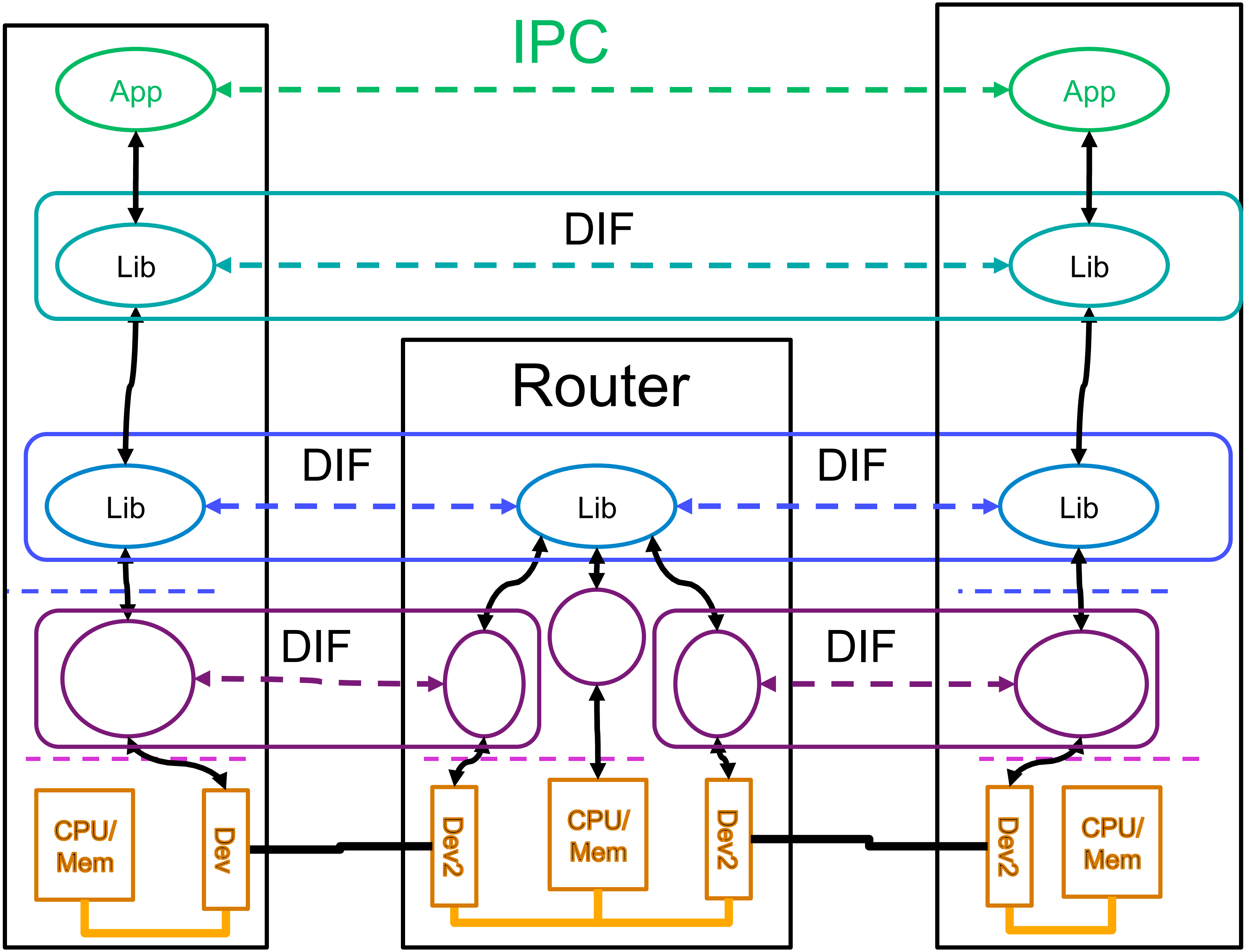
More layers



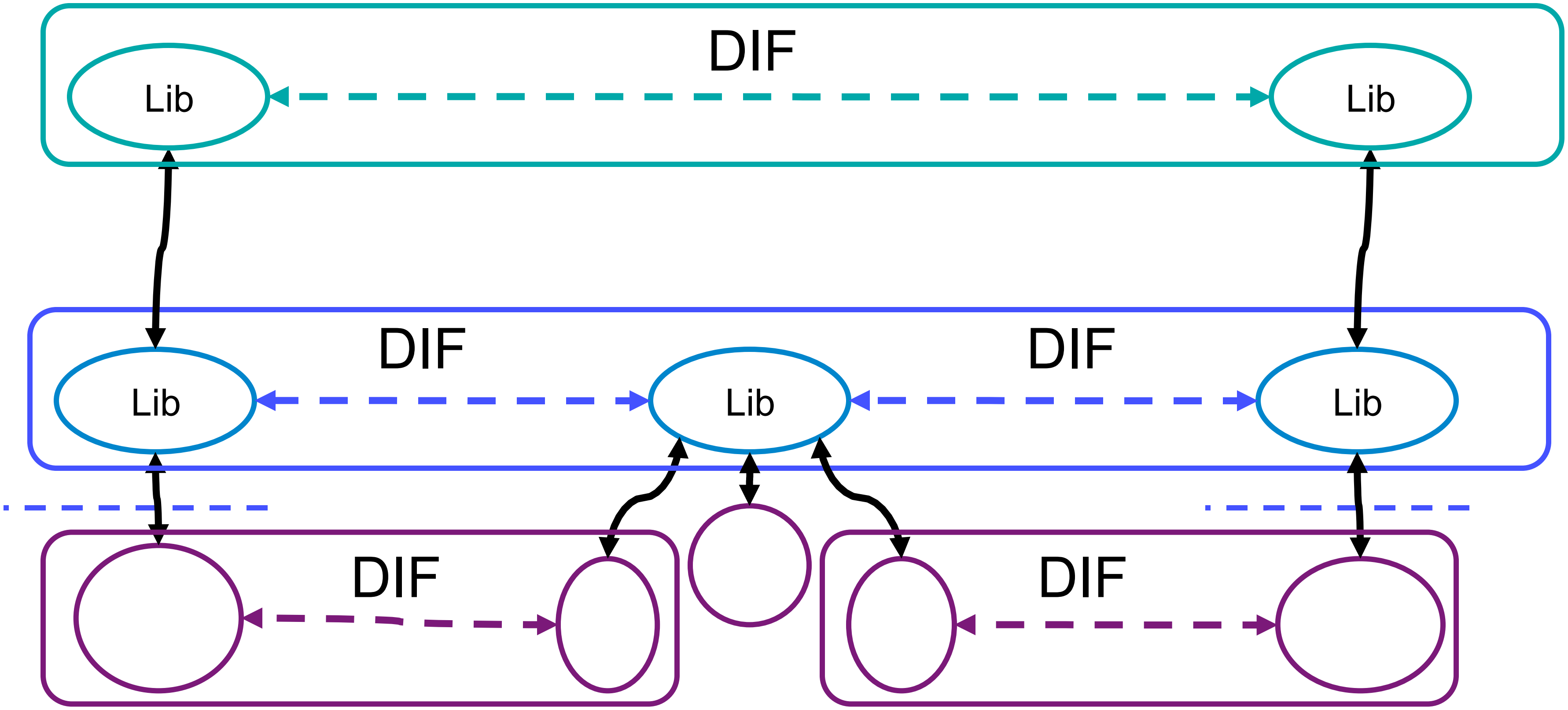
Different scopes

DIF= **D**istributed **I**PC **F**acility


Host
Host
IPC
App
App
Router
DIF
DIF
Lib
Lib
Lib
DIF
DIF
CPU/ Mem
Dev
Dev2
CPU/ Mem
Dev2
Dev2
CPU/ Mem


IPC
App
App
DIF
Lib
Lib
Router
DIF
DIF
Lib
Lib
Lib
DIF
DIF
CPU/ Mem
Dev
Dev2
CPU/ Mem
Dev2
Dev2
CPU/ Mem

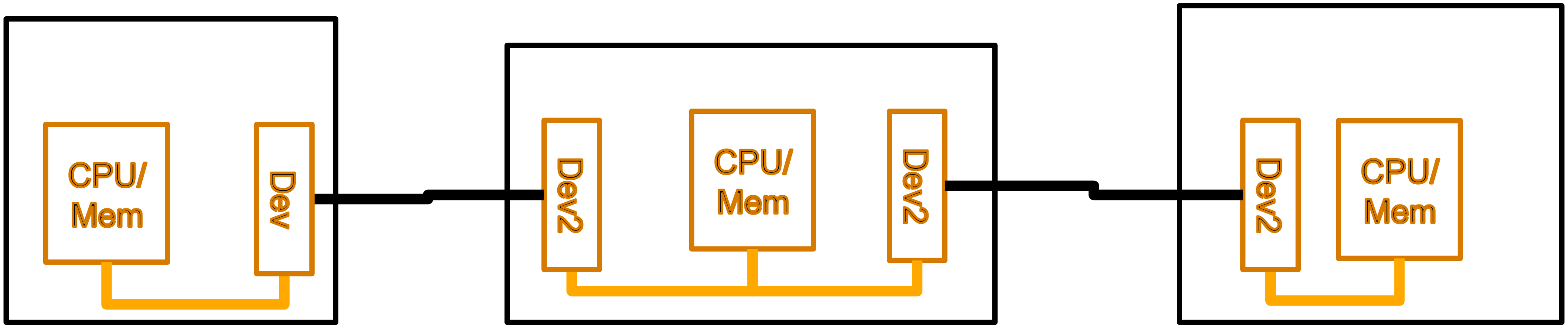
How many layers?

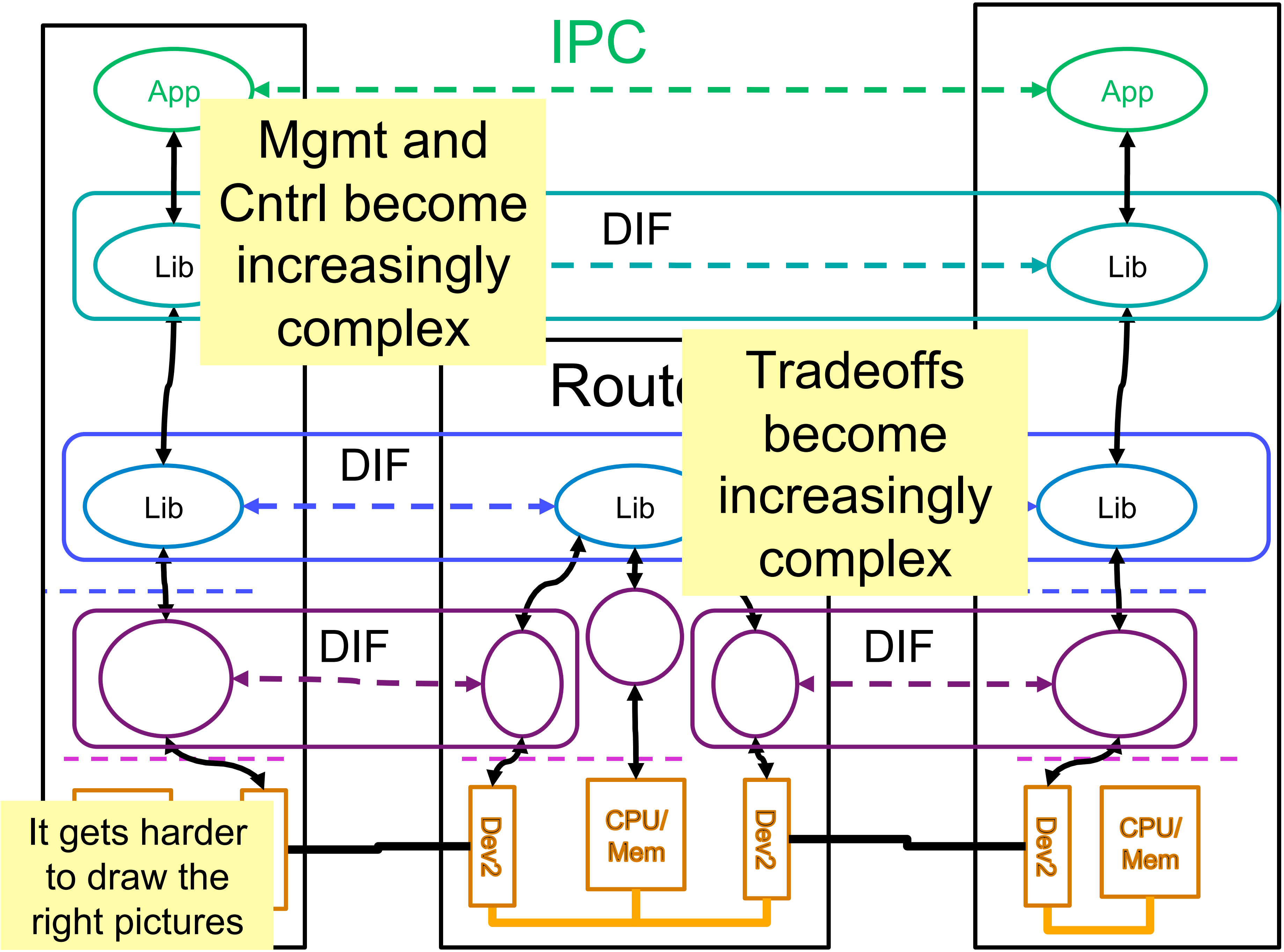


How many layers?

As many as you need to map distribute applications

Onto distributed resources

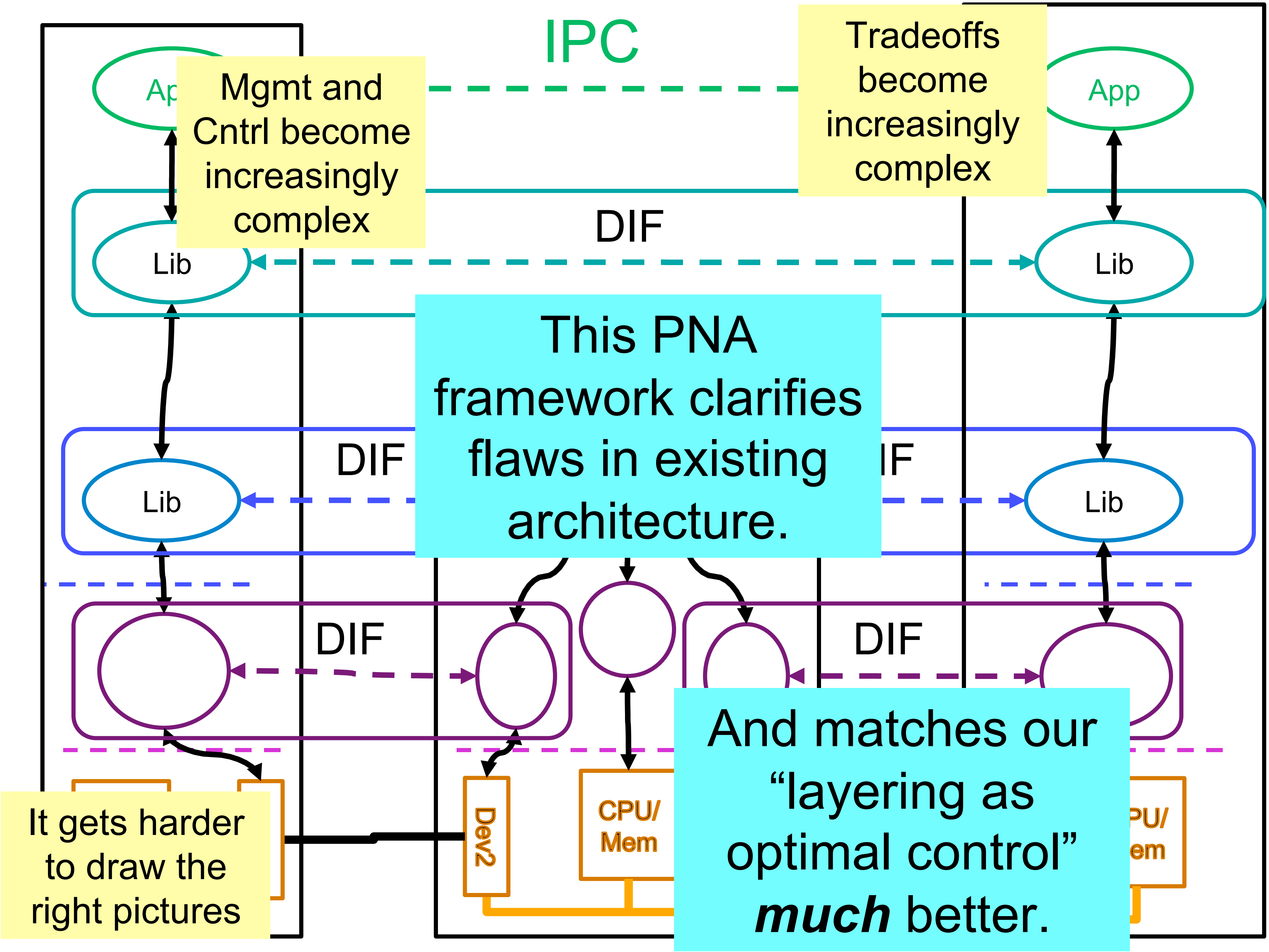




Mgmt and Cntrl become increasingly complex

Tradeoffs become increasingly complex

It gets harder to draw the right pictures



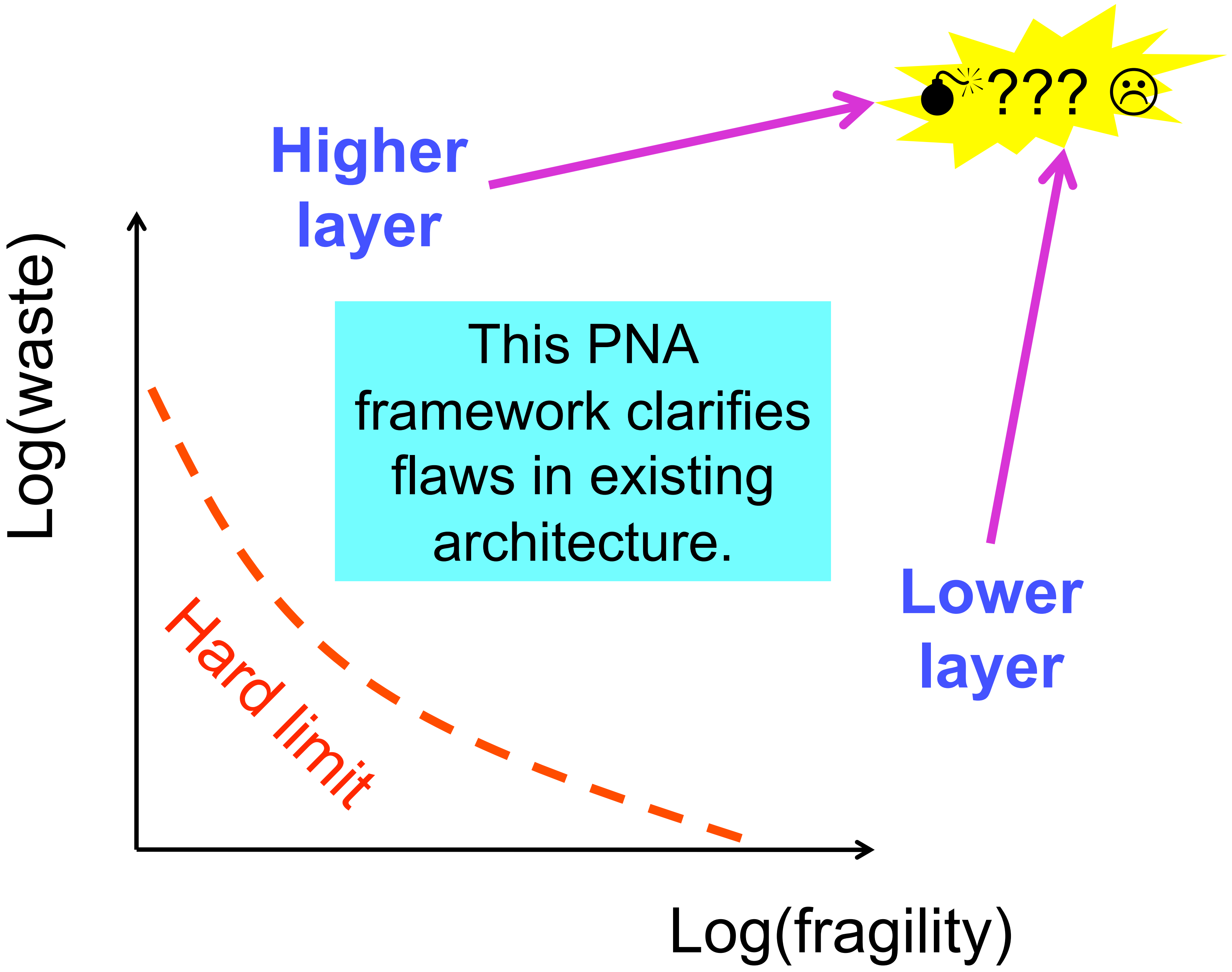
Mgmt and Cntrl become increasingly complex

Tradeoffs become increasingly complex

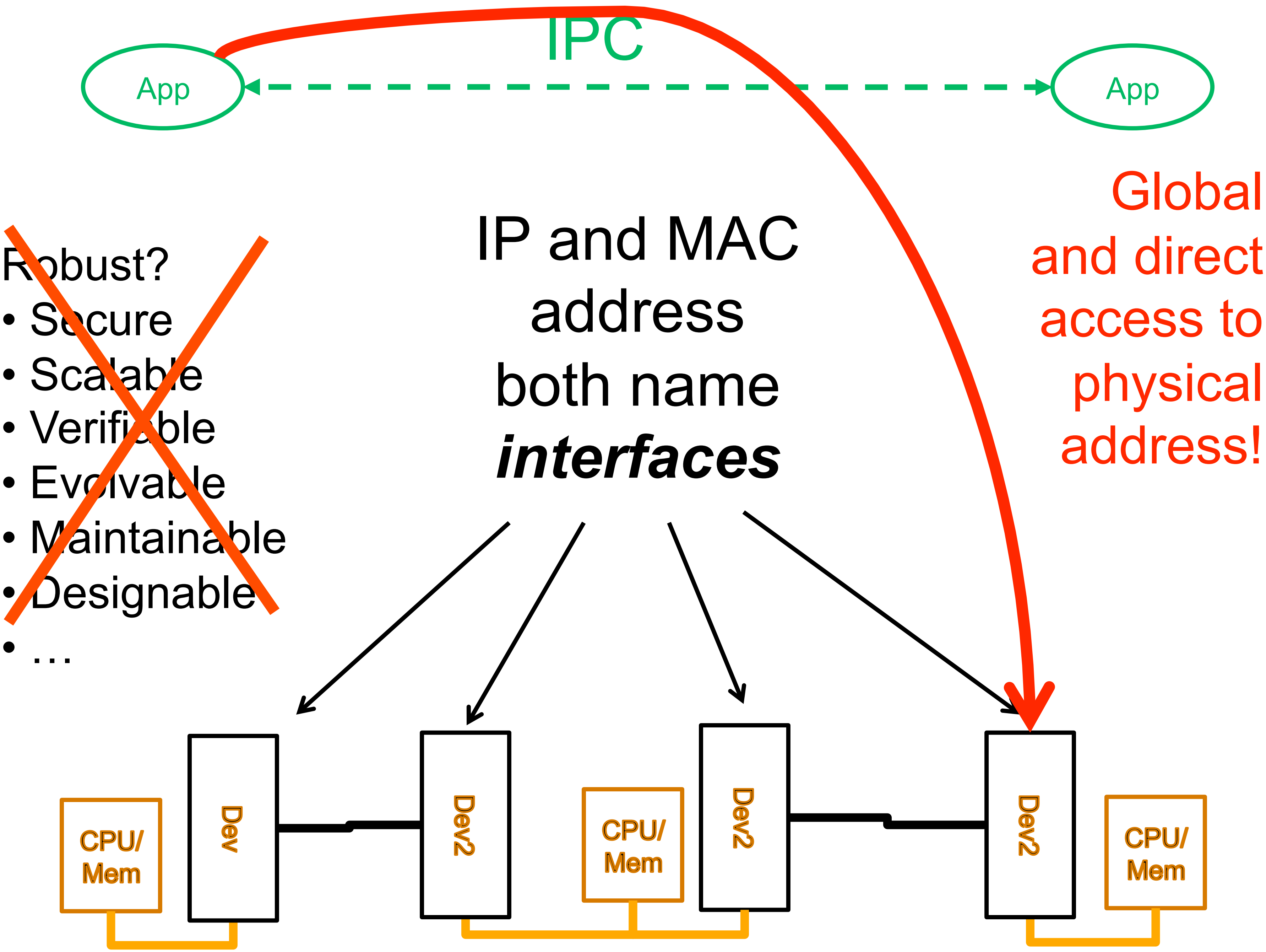
This PNA framework clarifies flaws in existing architecture.

And matches our “layering as optimal control” ***much*** better.

It gets harder to draw the right pictures



This PNA framework clarifies flaws in existing architecture.



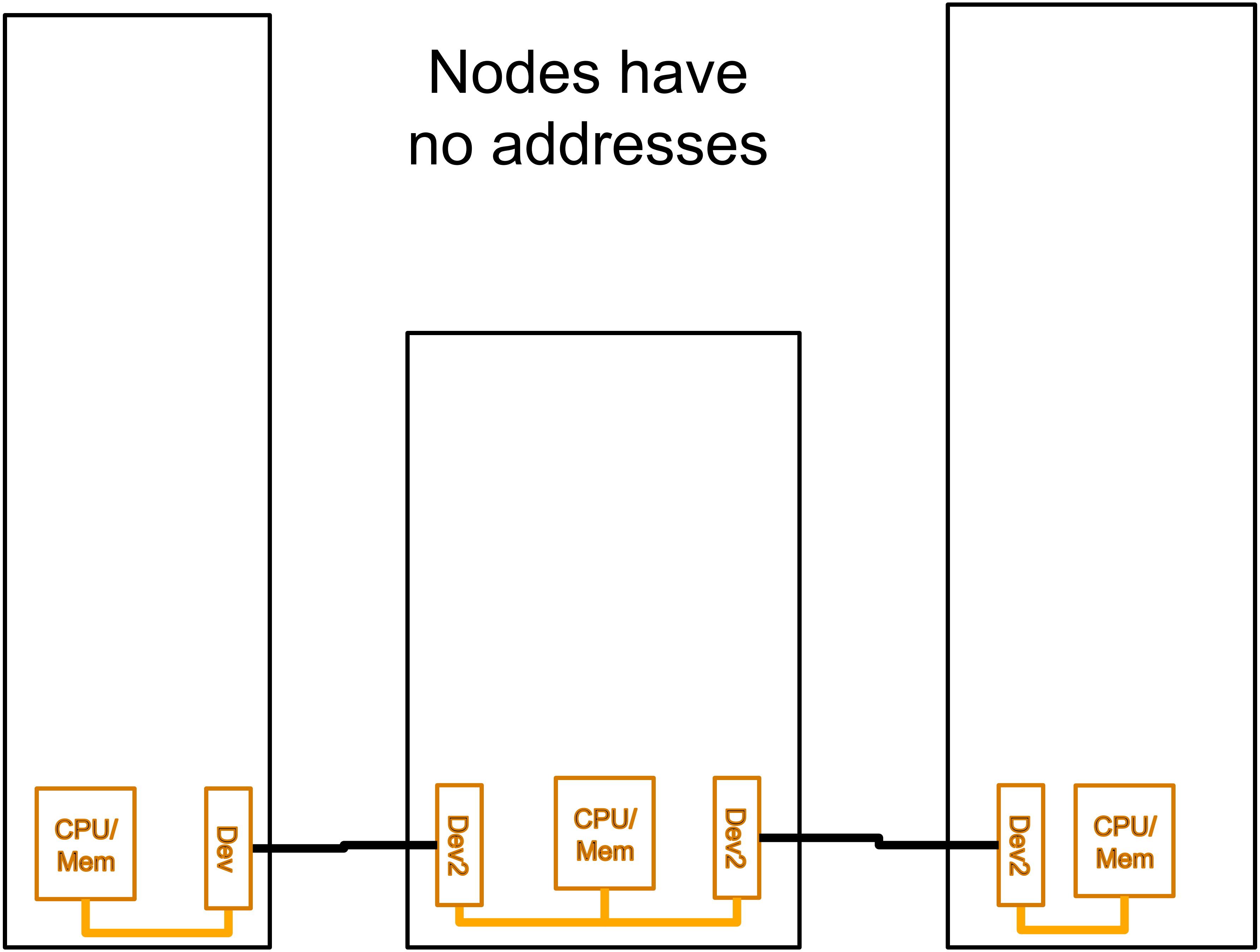
~~Robust?~~

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

IP and MAC address both name ***interfaces***

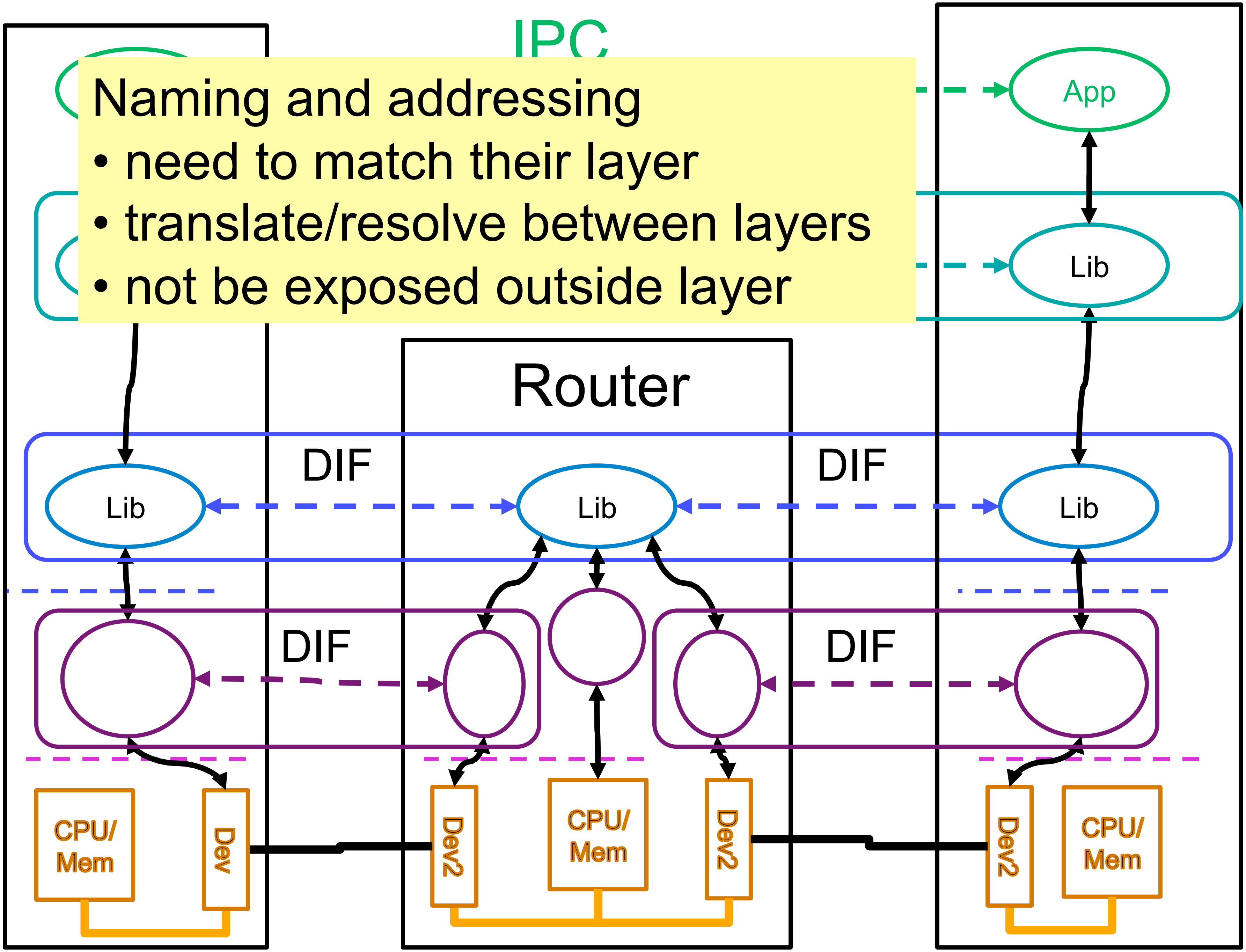
Global and direct access to physical address!

Nodes have no addresses



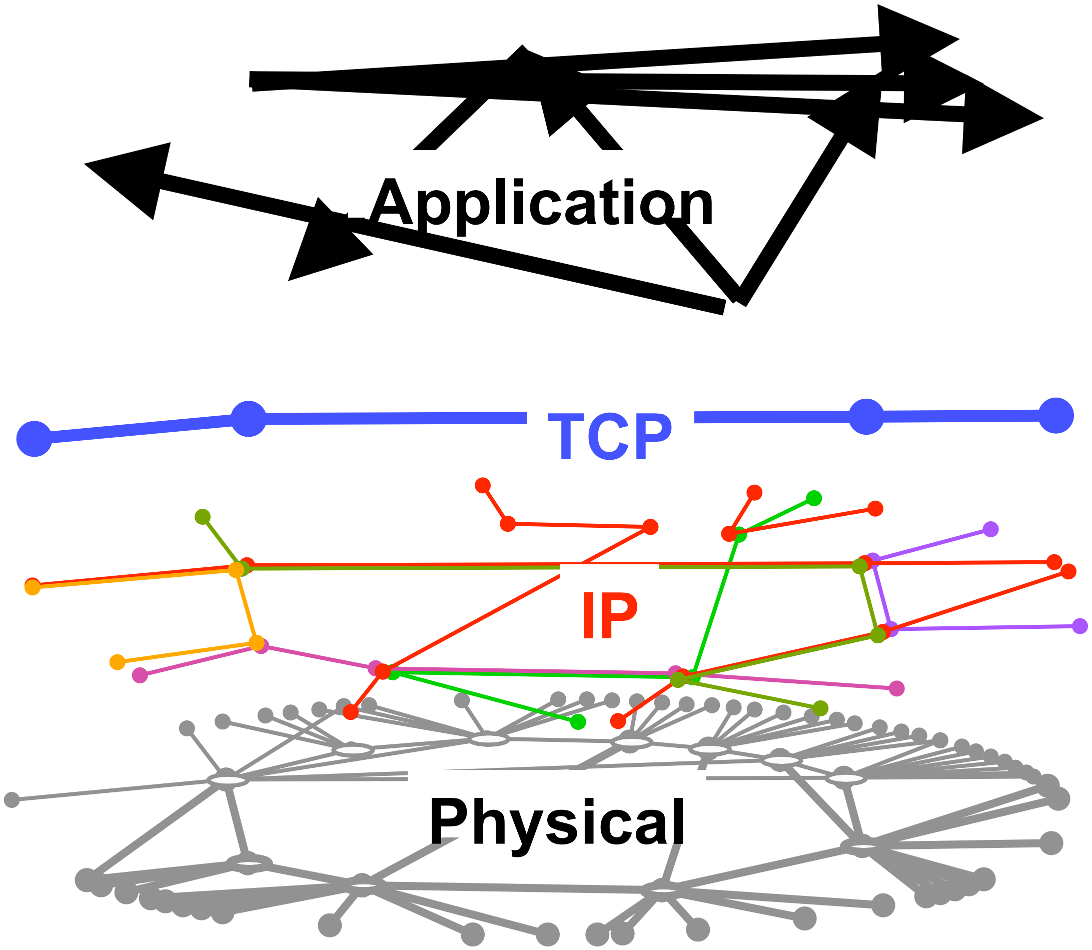
Naming and addressing

- need to match their layer
- translate/resolve between layers
- not be exposed outside layer



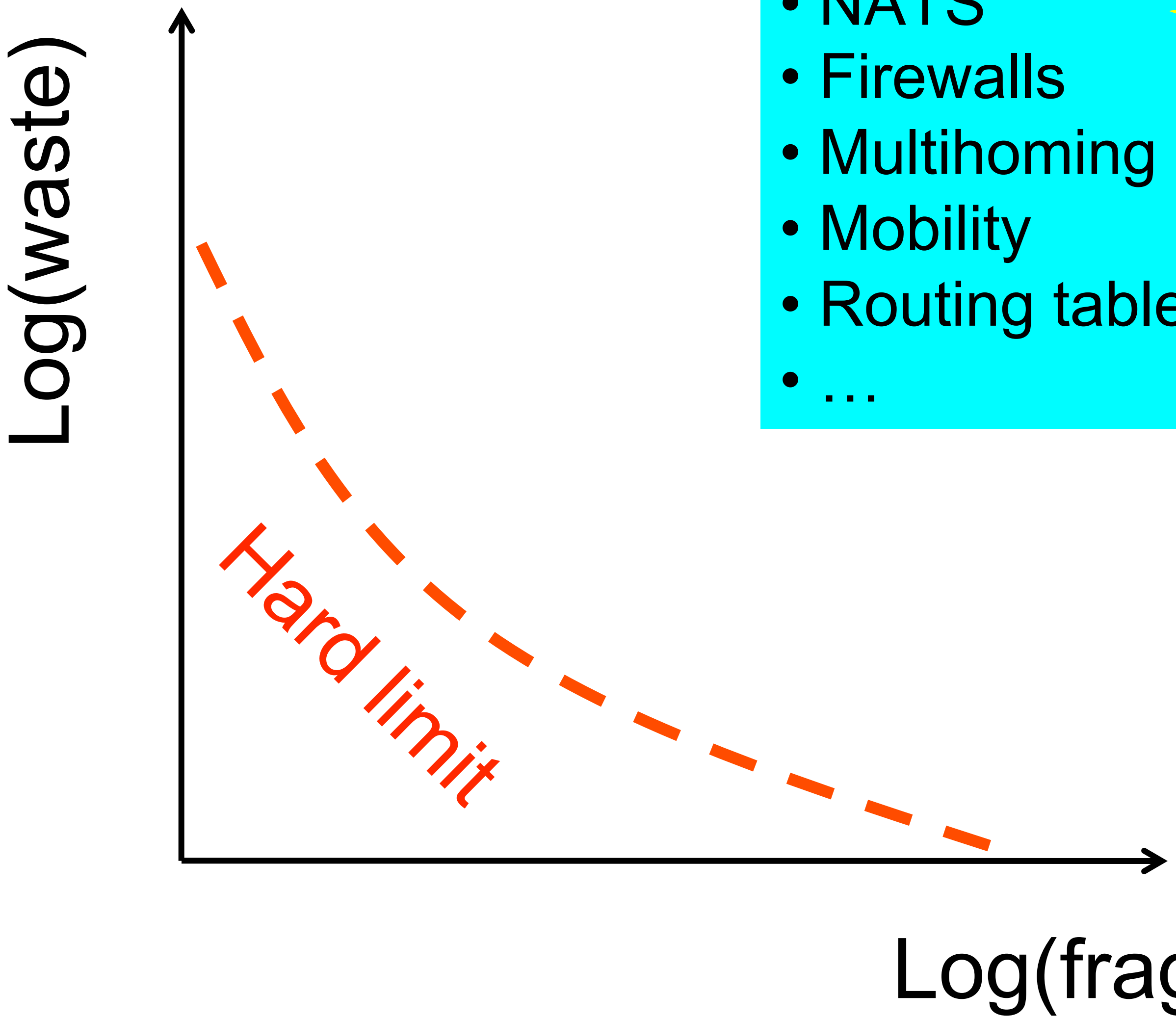
Naming and addressing
- need to match their layer
- translate/resolve between layers
- not be exposed outside layer

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



Architecture issues
- DNS
- NATS
- Firewalls
- Multihoming
- Mobility
- Routing table size
- …

💣??? ☹

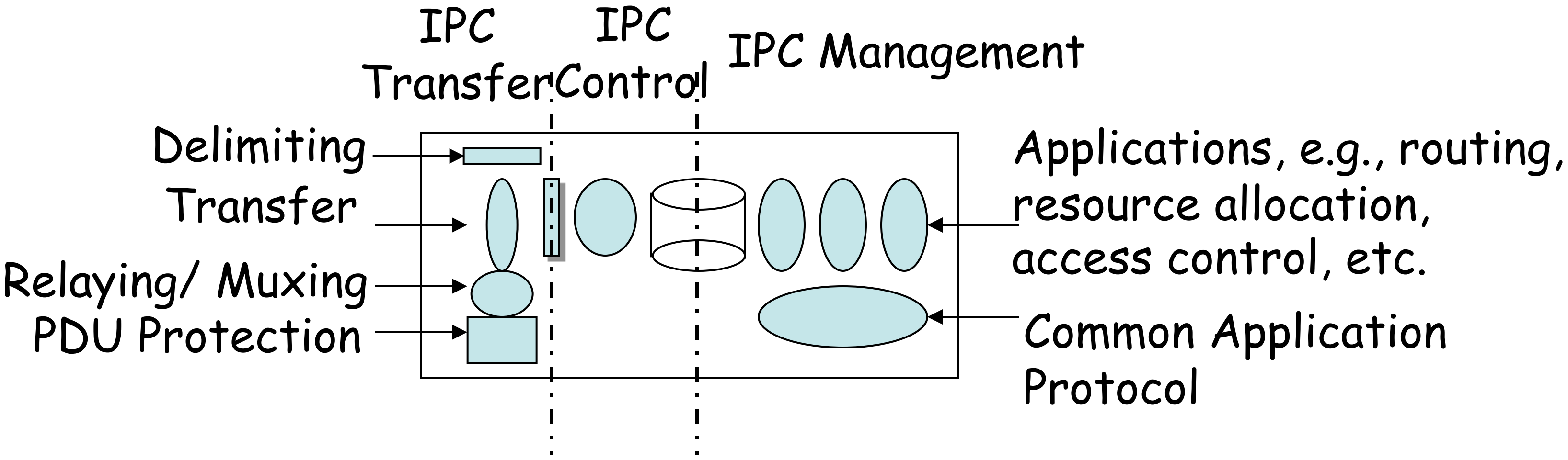


Next steps?

- New course this term? (CDS 213?)
- Discuss at 1pm
- Flesh out details
- Integrated control, comms, computing, thermo/statmech, optimization, games, etc
- Motivated by very generic network challenges

Next steps?

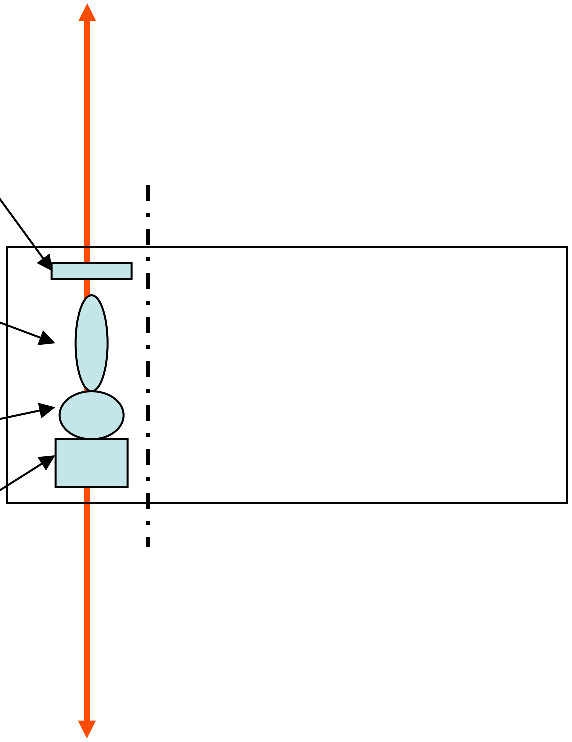
Start with this picture from PNA



And categorize these

- Delimiting
- Initial State Synch
- Policy Selection
- Addressing
- Flow/Connection Identifier
- Relaying
- Multiplexing
- Ordering
- Frag./Reassembly
- Combining/Separation
- Data Corruption
- Lost /Duplicate Detection
- Flow Control
- Forward Error Cor.
- Ack/Retran Control
- Compression
- Authentication
- Access Control
- Integrity
- Confidentiality
- Nonrepudiation
- Activity

IPC Xfer



Delimiting

Transfer

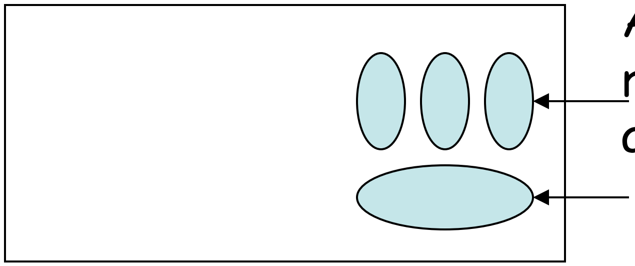
Addressing
Ordering
Frag./Reassembly
Combining/Separation
Lost /Duplicate Detection

Relaying
Multiplexing

SDU Protection

Data Corruption
Integrity
Confidentiality
Compression

IPC Mgmt

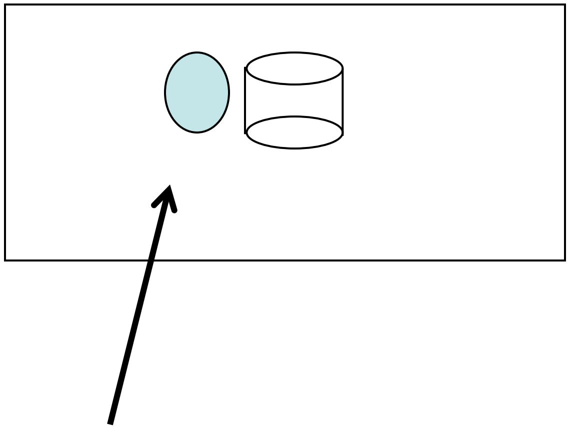


Applications, e.g., routing, resource allocation, access control, etc.

Common Application Protocol

Routing
Policy Selection
Flow/Connection Identifier
Access Control

IPC
Cntrl



Flow Control
Ack/Retran Control

summary

IPC Xfer

Delimiting

Addressing
Ordering
Frag./Reassembly
Combining/Separation
Lost /Duplicate Detection

Relaying
Multiplexing

SDU Protection

Data Corruption?
Integrity
Confidentiality
Compression

IPC Cntrl

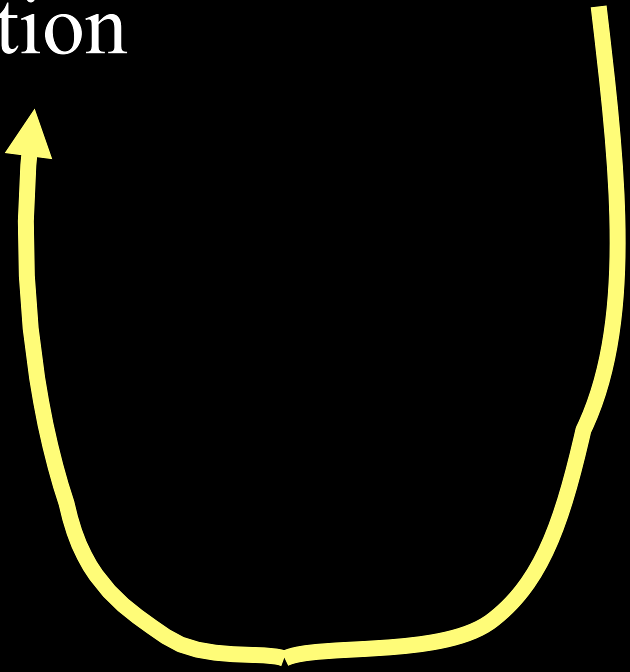
Flow Control
Ack/Retran Control

IPC Mgmt

Policy Selection
Flow/Connection Identifier
Access Control

Data Corruption
TTL
Forward Error Cor.

In the real (vs virtual) world



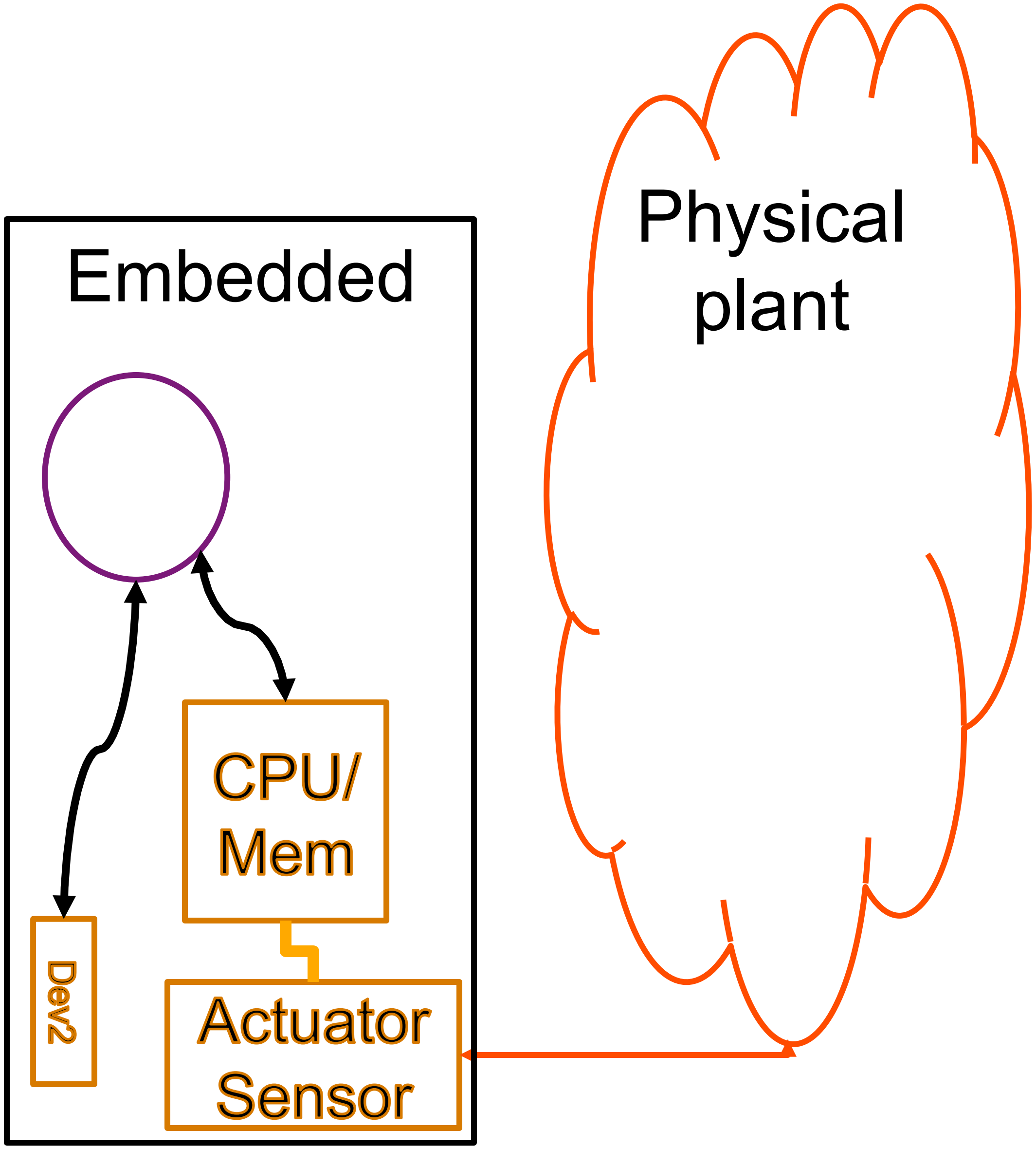
What matters:

- Action

What doesn't:

- Data
- Information
- Computation
- Learning
- Decision
- …

Embedded



Physical
plant

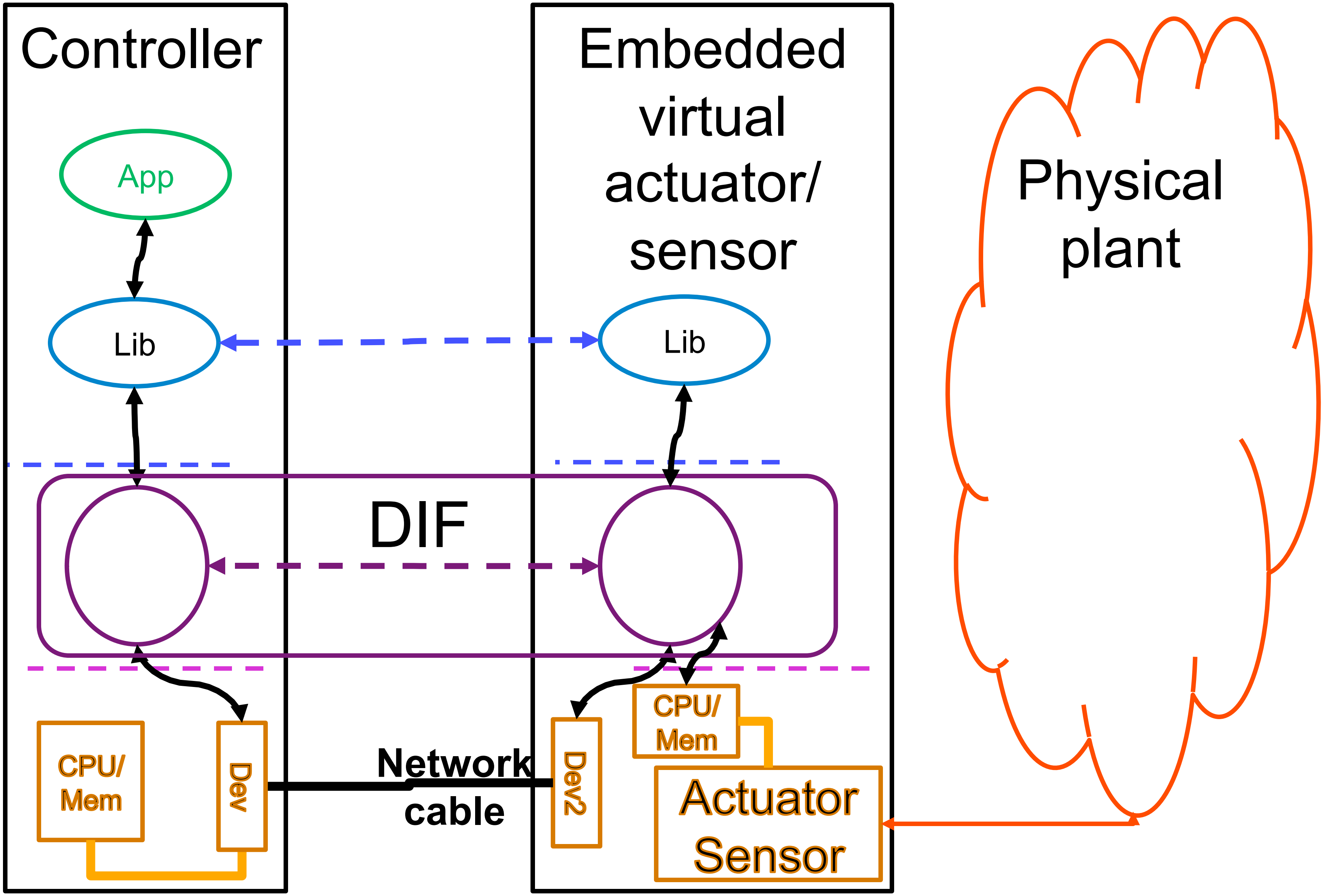
Embedded

CPU/
Mem

Dev2

Actuator
Sensor

Networked embedded



Controller

App

Lib

DIF

CPU/
Mem

Dev

Network
cable

Embedded
virtual
actuator/
sensor

Lib

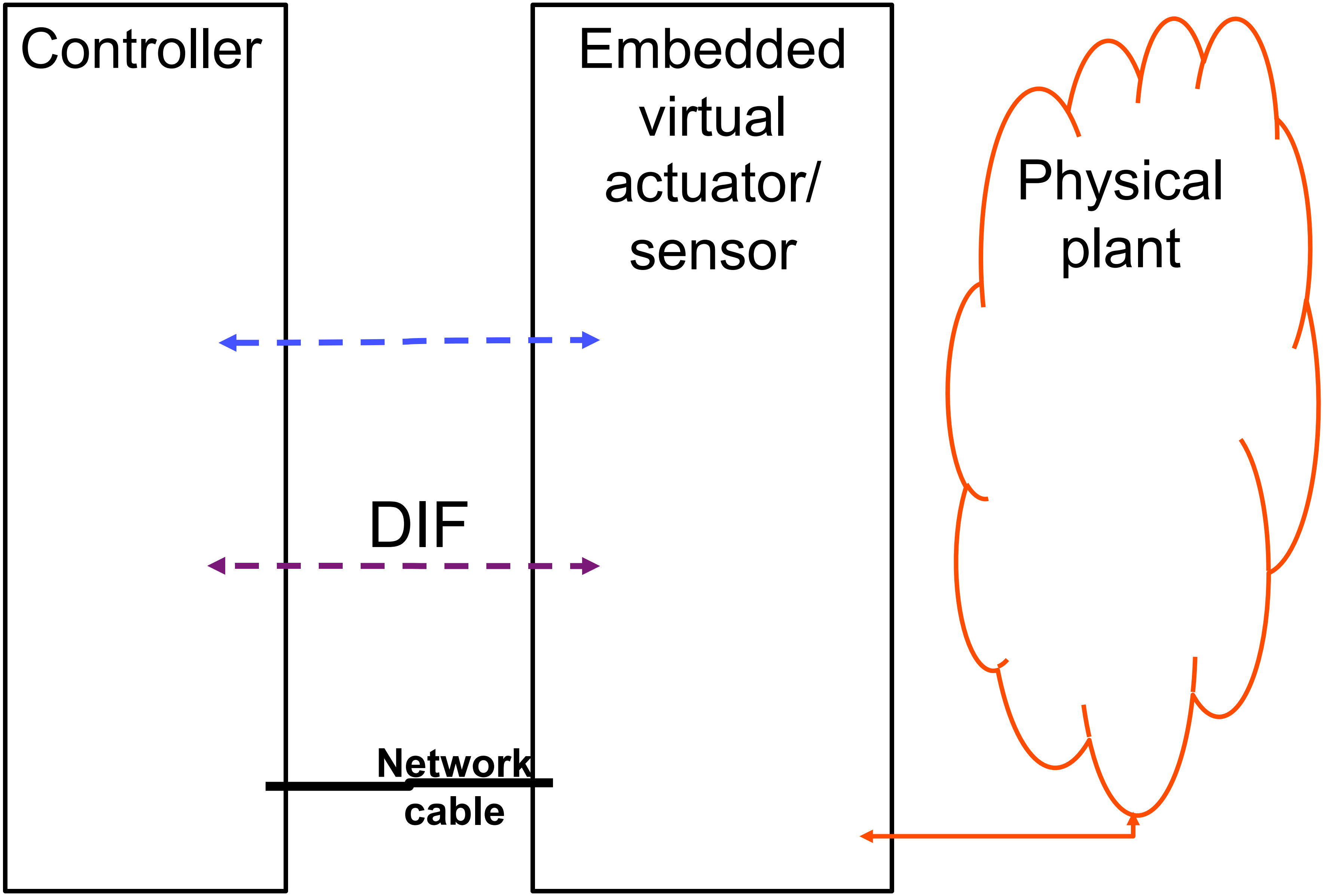
Dev2

CPU/
Mem

Actuator
Sensor

Physical
plant

Meta-layering of cyber-phys control



Controller

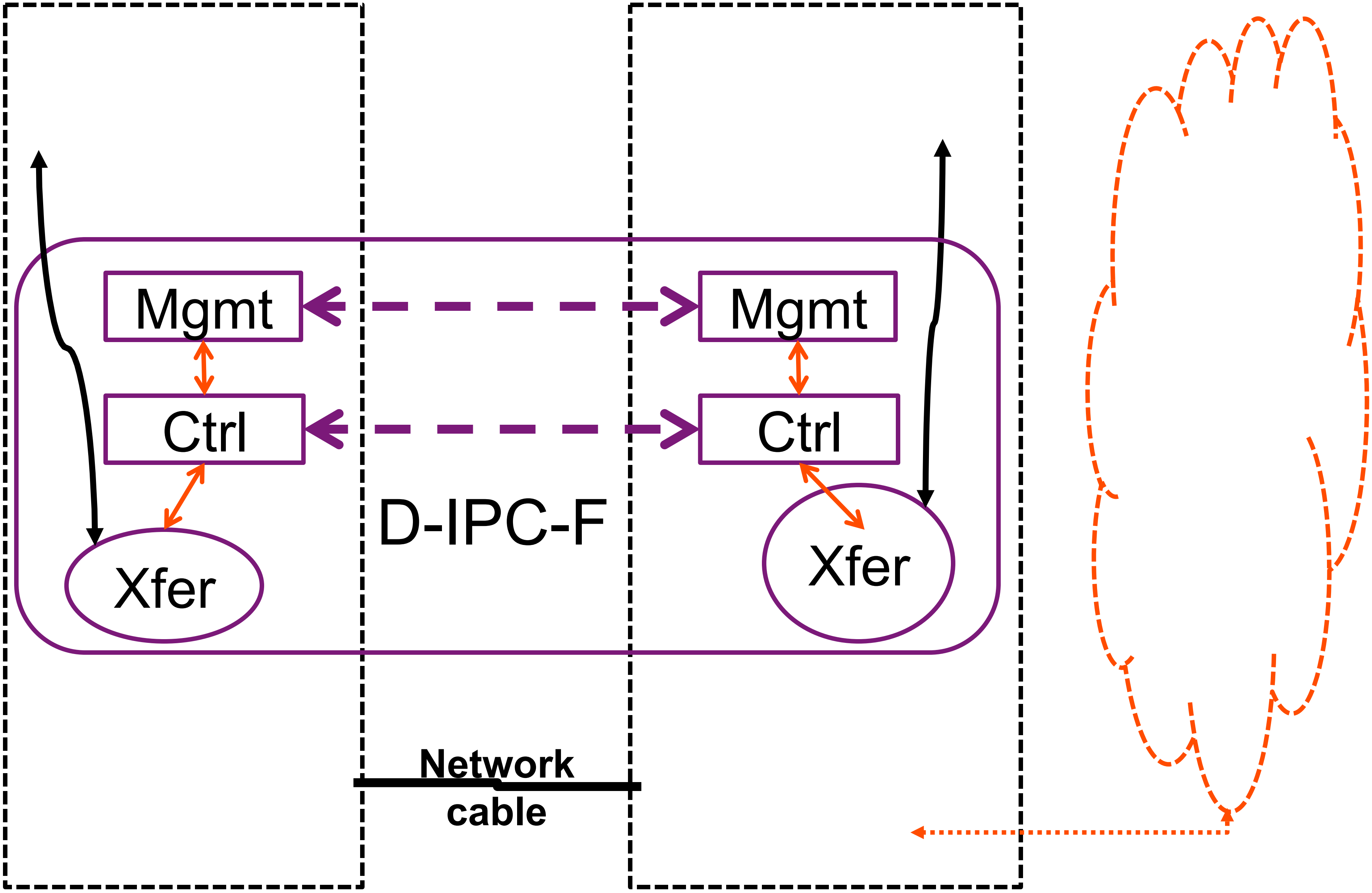
Embedded virtual actuator/ sensor

Physical plant

DIF

Network cable

Micro-layering of D-IPC-F



Biology versus the Internet

Similarities

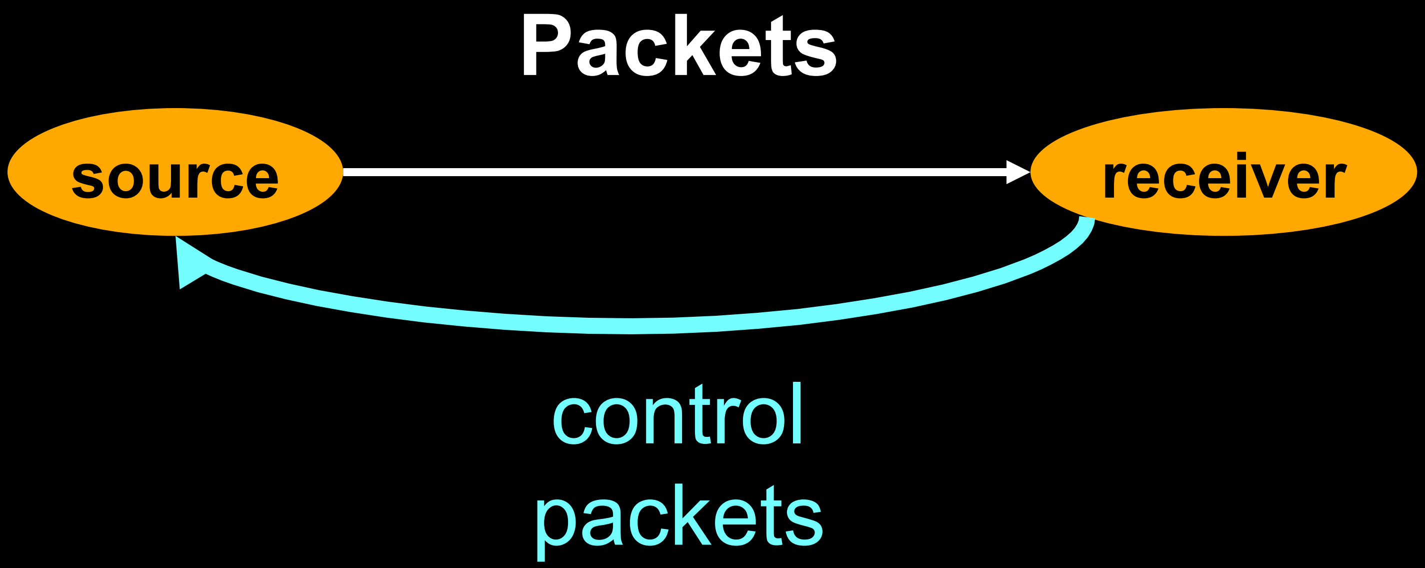
- Evolvable architecture
- **Robust yet fragile**
- **Constraints/deconstrain**
- **Layering, modularity**
- **Hourglass with bowties**
- Feedback
- Dynamics
- Distributed/decentralized
- *Not* scale-free, edge-of-chaos, self-organized criticality, etc

Differences

- Metabolism
- Materials and energy
- **Autocatalytic feedback**
- Feedback complexity
- Development and regeneration
- >4B years of evolution

Focus on
bacterial biosphere

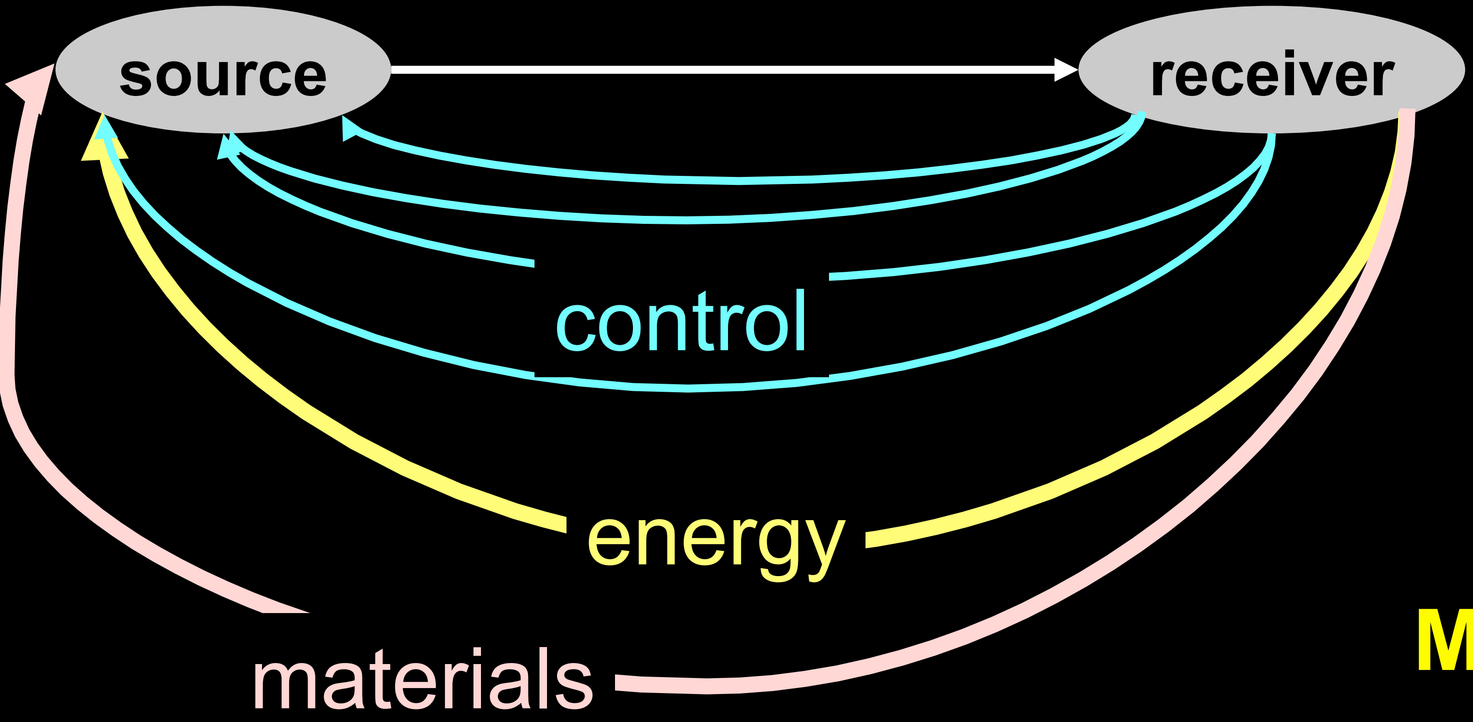
Control of the Internet



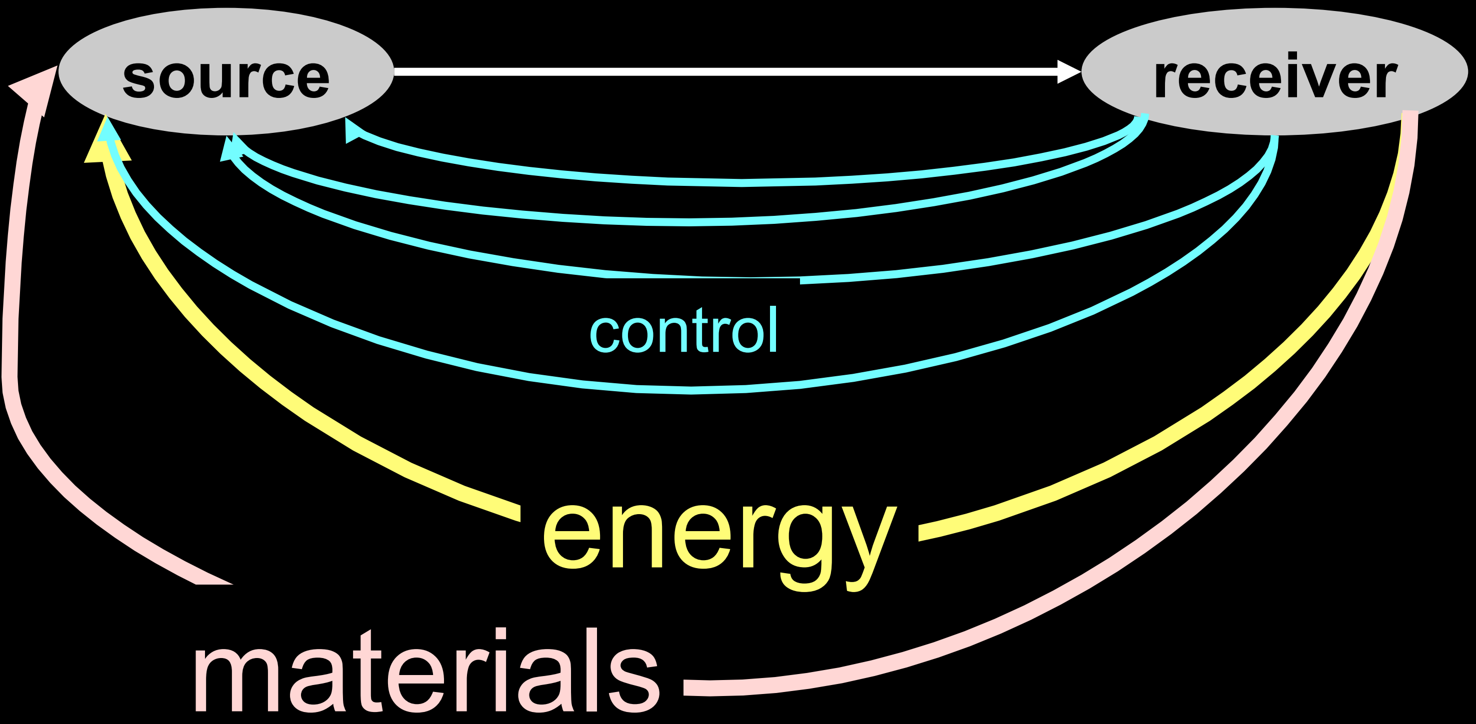
signaling
gene expression
metabolism
lineage

Biological pathways

signaling
gene expression
metabolism
lineage

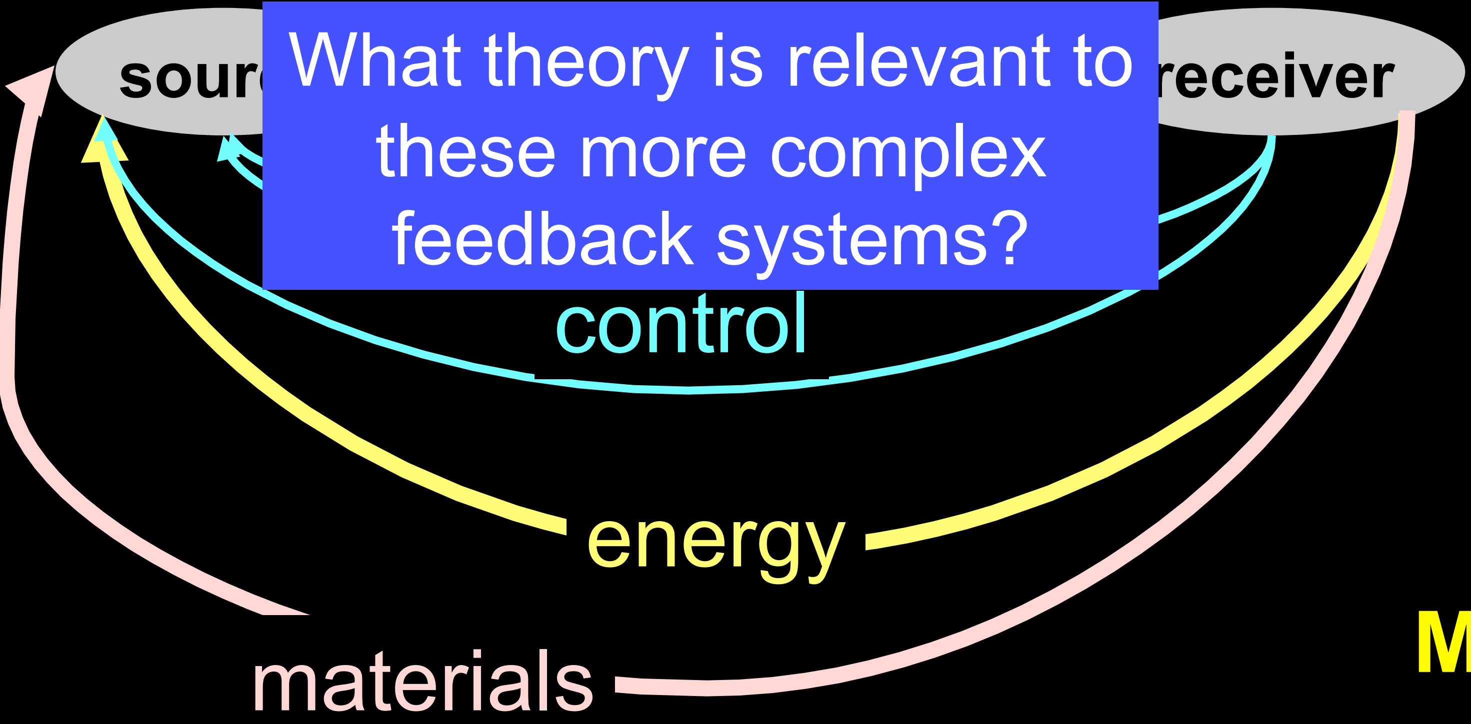


More complex feedback



Autocatalytic feedback

signaling
gene expression
metabolism
lineage

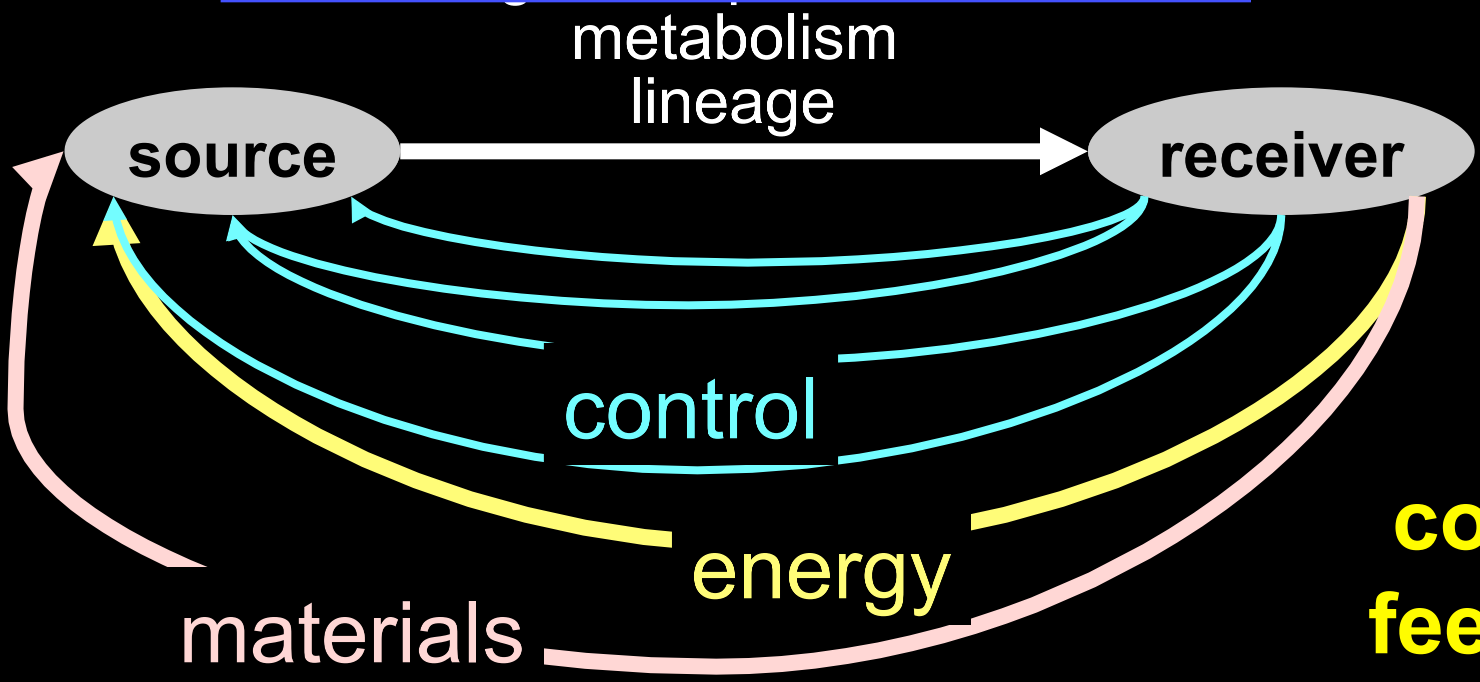


What theory is relevant to these more complex feedback systems?

More complex feedback

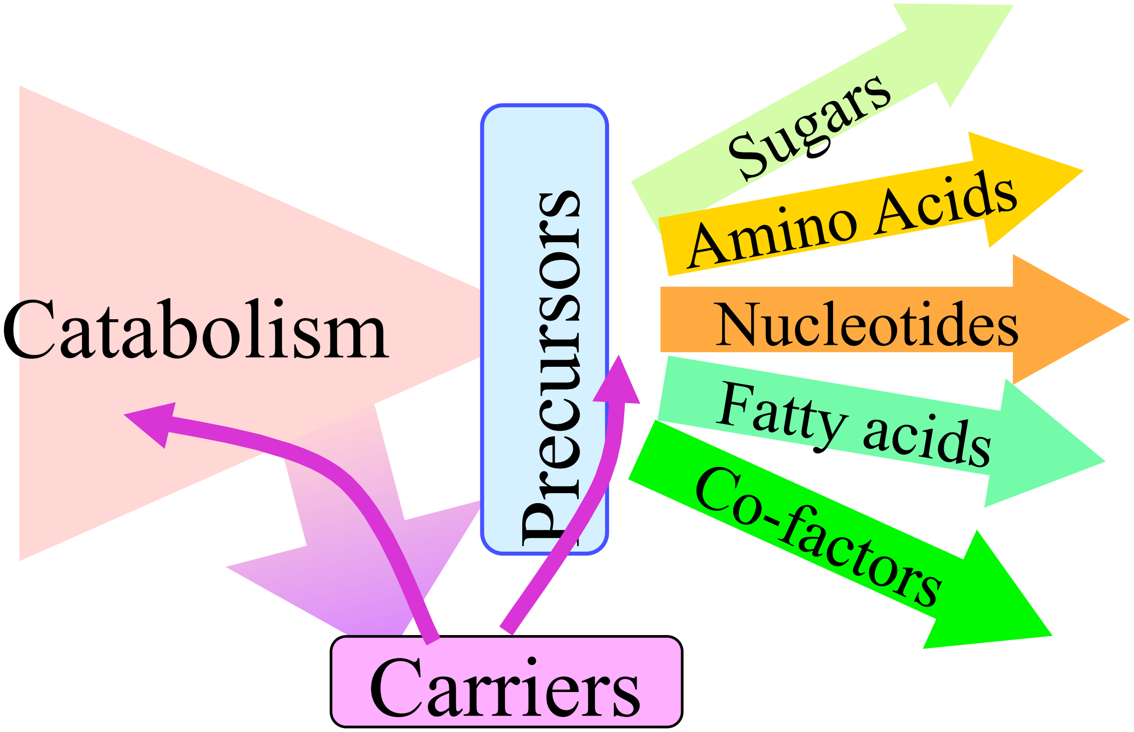
What theory is relevant to these more complex feedback systems?

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



More complex feedback

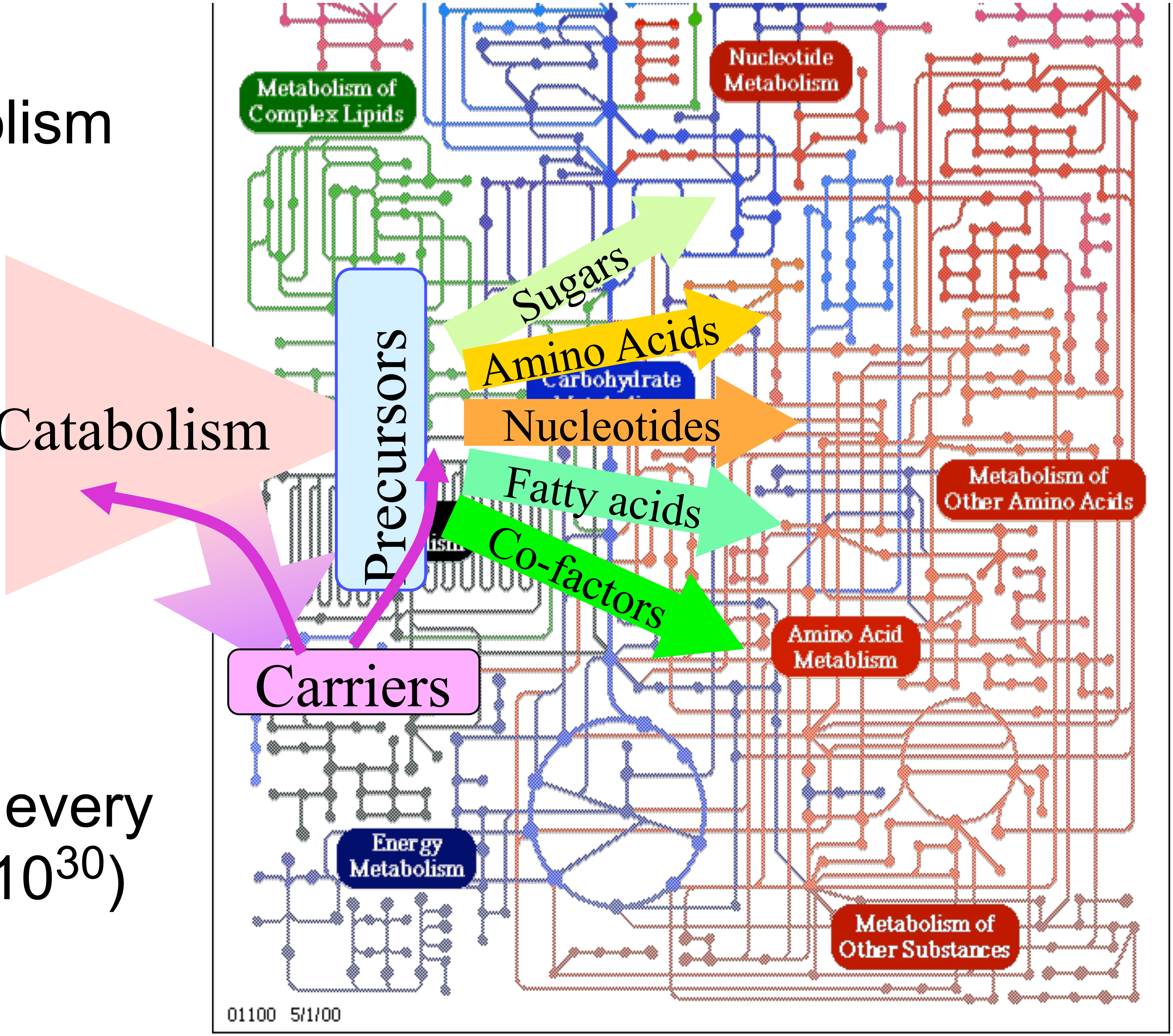
Inside every cell



Core metabolic bowtie

Skipping the “OS” story, right to networks

Core metabolism



Catabolism

Precursors

Carriers

Sugars

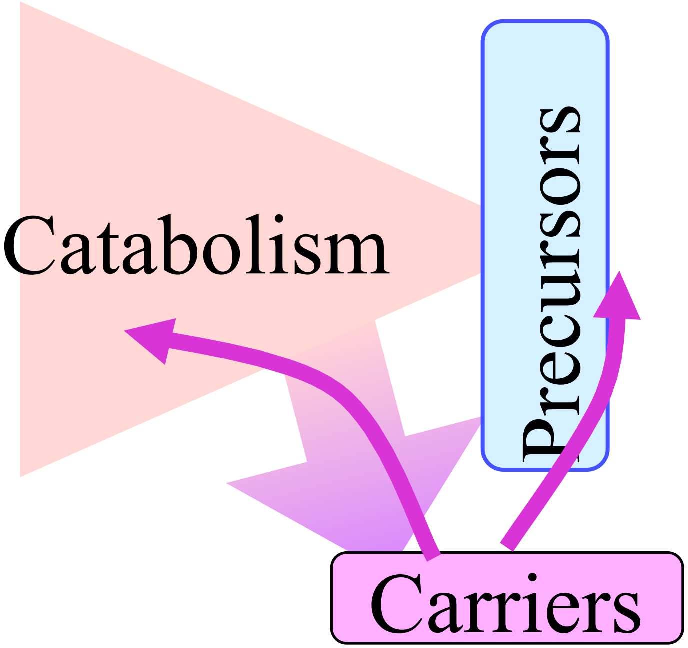
Amino Acids

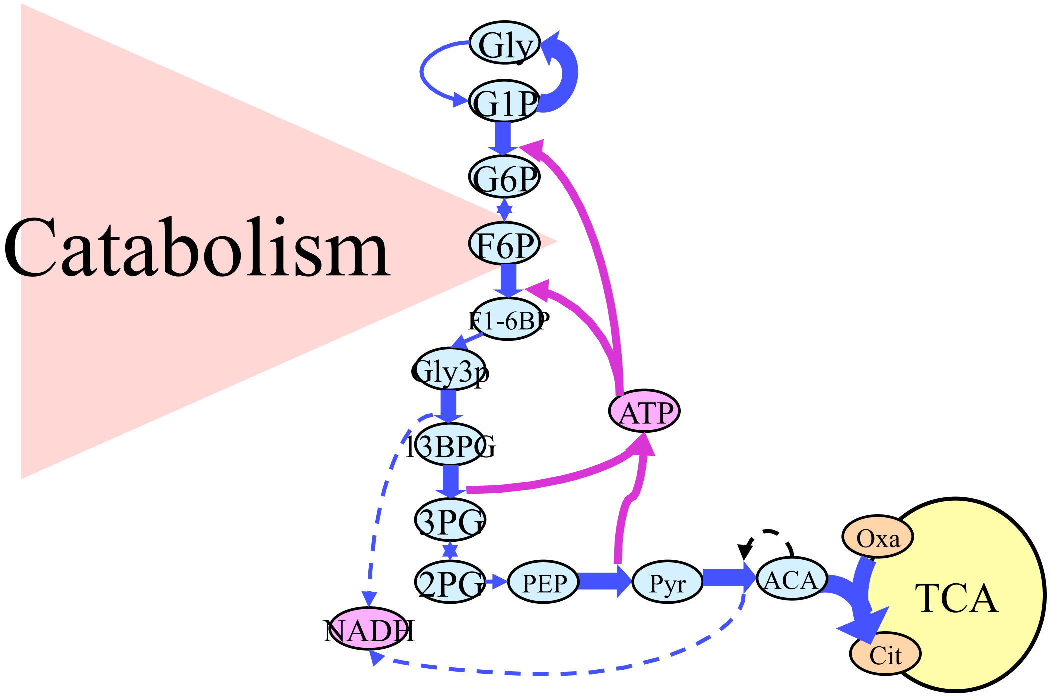
Nucleotides

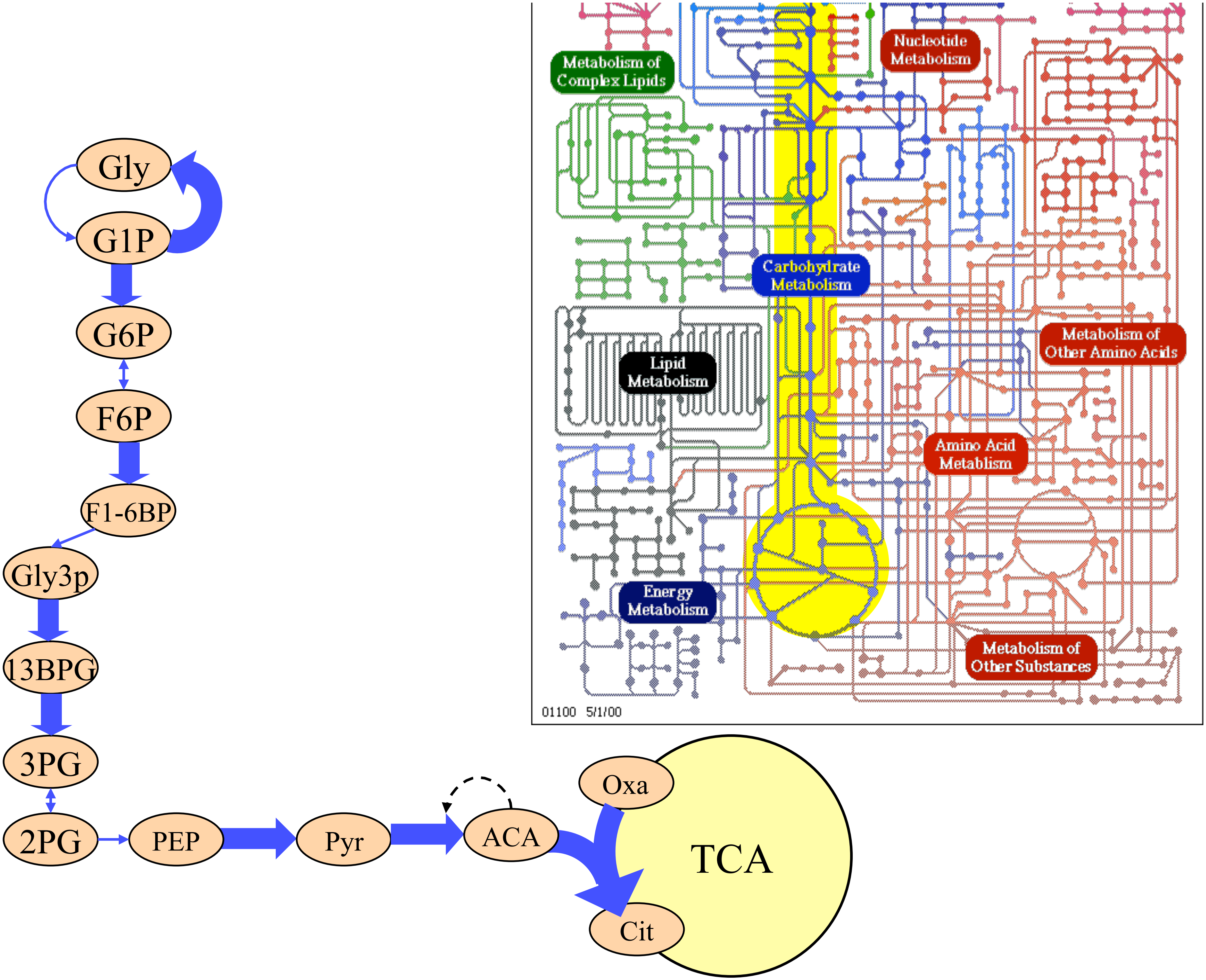
Fatty acids

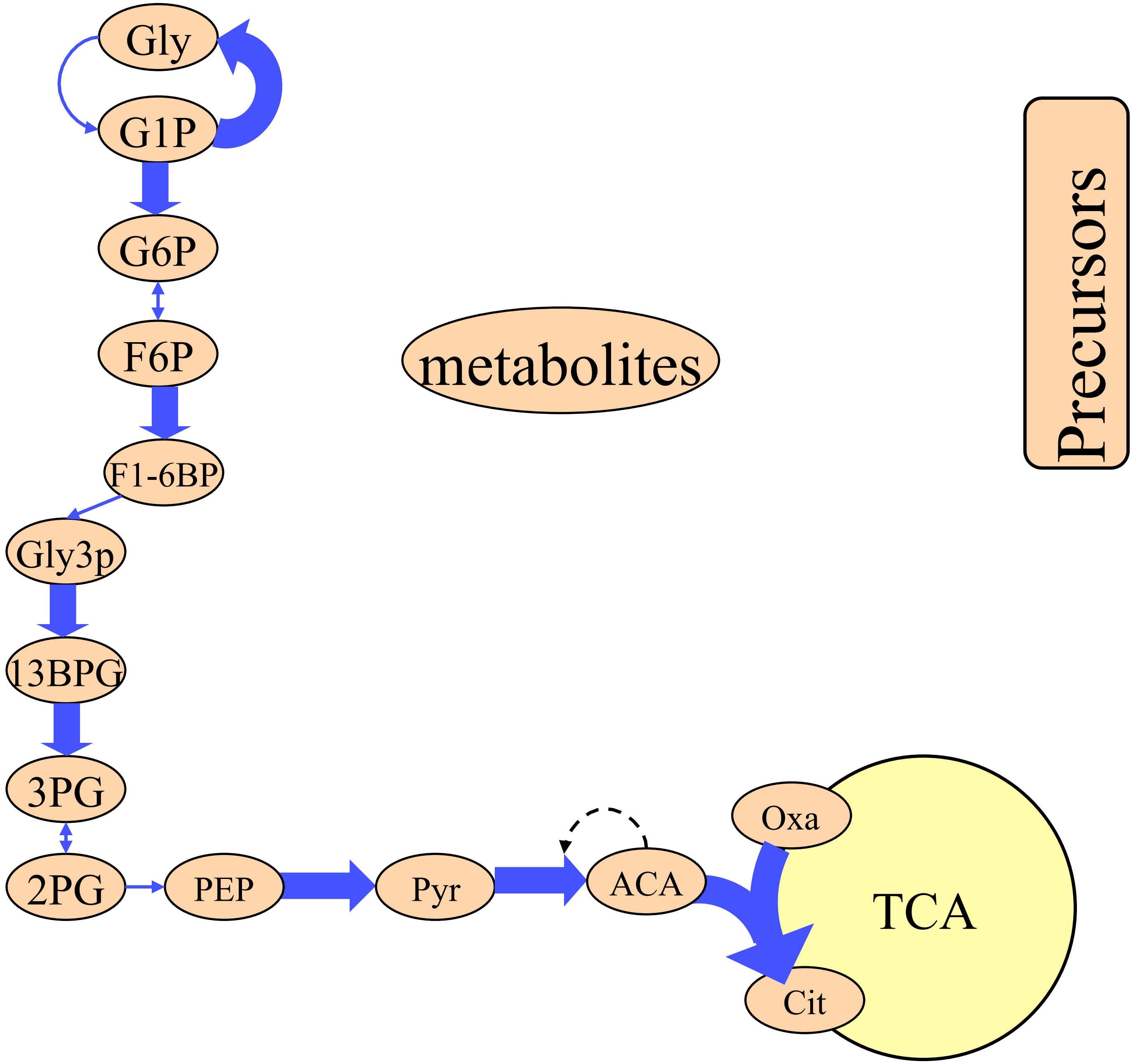
Co-factors

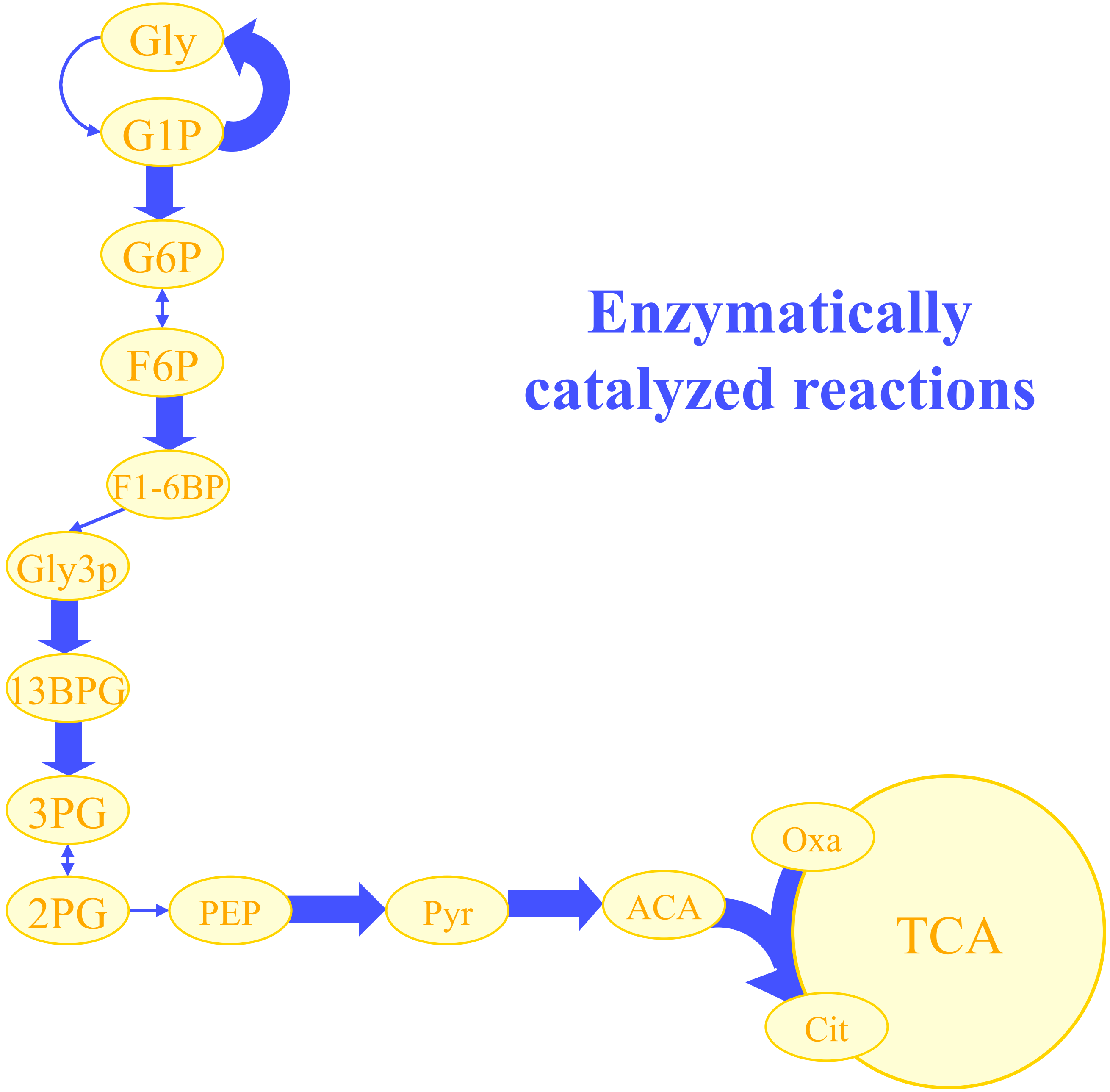
Inside every cell ($\approx 10^{30}$)


Catabolism
Precursors
Carriers


Catabolism
Gly
G1P
G6P
F6P
F1-6BP
Gly3p
13BPG
3PG
2PG
PEP
Pyr
ACA
ATP
NADH
Oxa
Cit
TCA

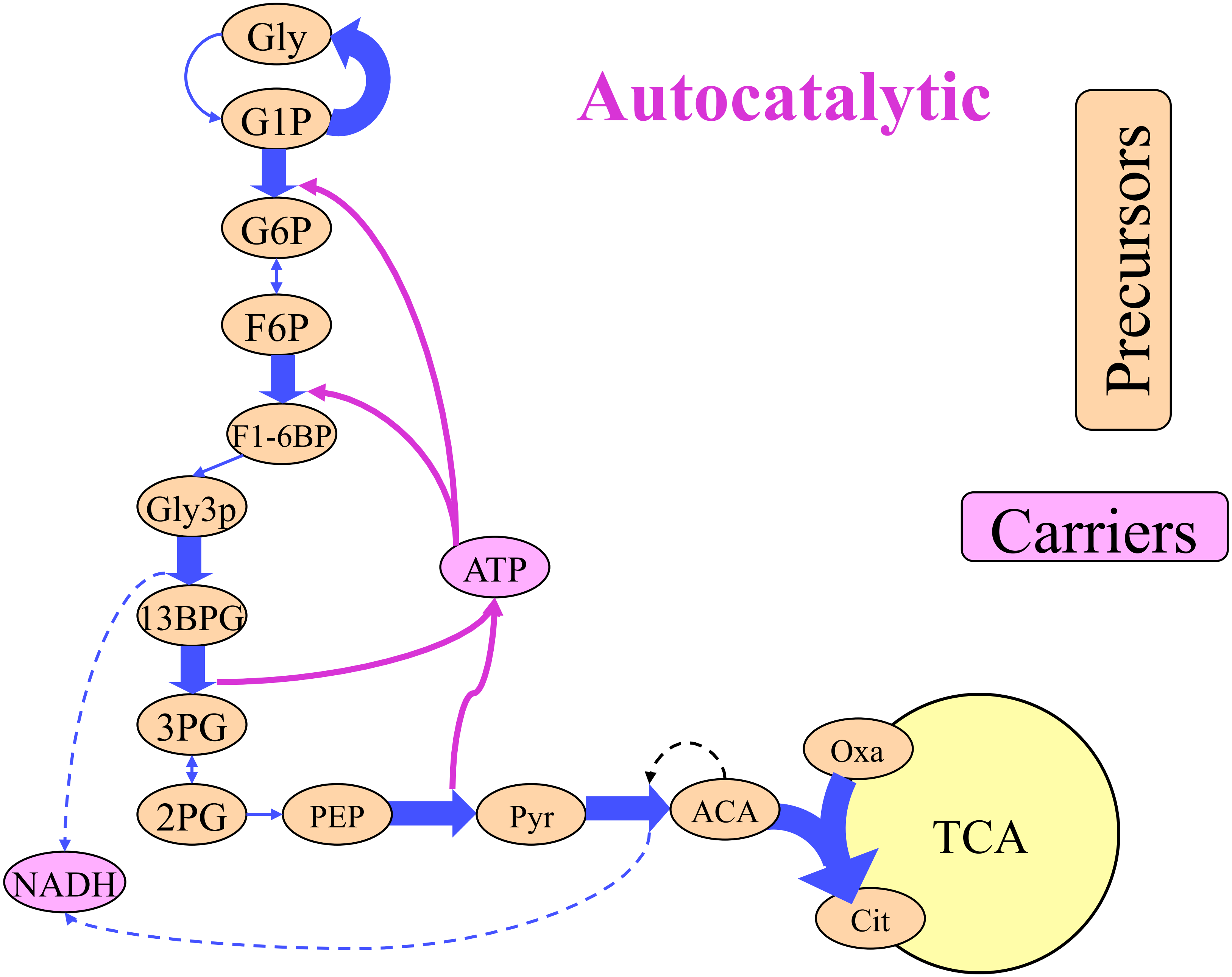

Gly
G1P
G6P
F6P
F1-6BP
Gly3p
13BPG
3PG
2PG
PEP
Pyr
ACA
Oxa
TCA
Cit
Metabolism of Complex Lipids
Nucleotide Metabolism
Carbohydrate Metabolism
Lipid Metabolism
Metabolism of Other Amino Acids
Amino Acid Metablism
Energy Metabolism
Metabolism of Other Substances
01100 5/1/00


Gly
G1P
G6P
F6P
F1-6BP
Gly3p
13BPG
3PG
2PG
PEP
Pyr
ACA
Oxa
Cit
TCA
metabolites
Precursors

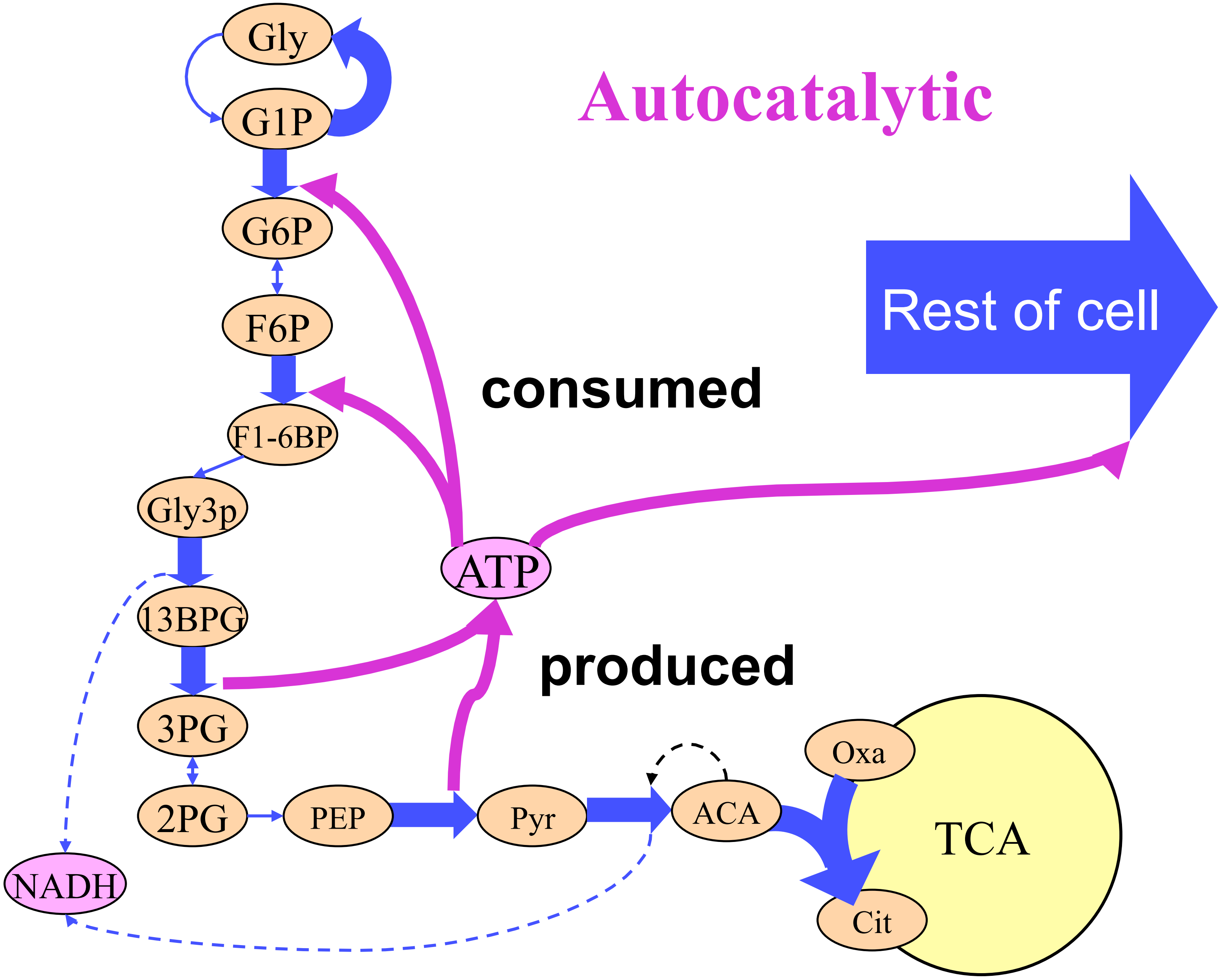


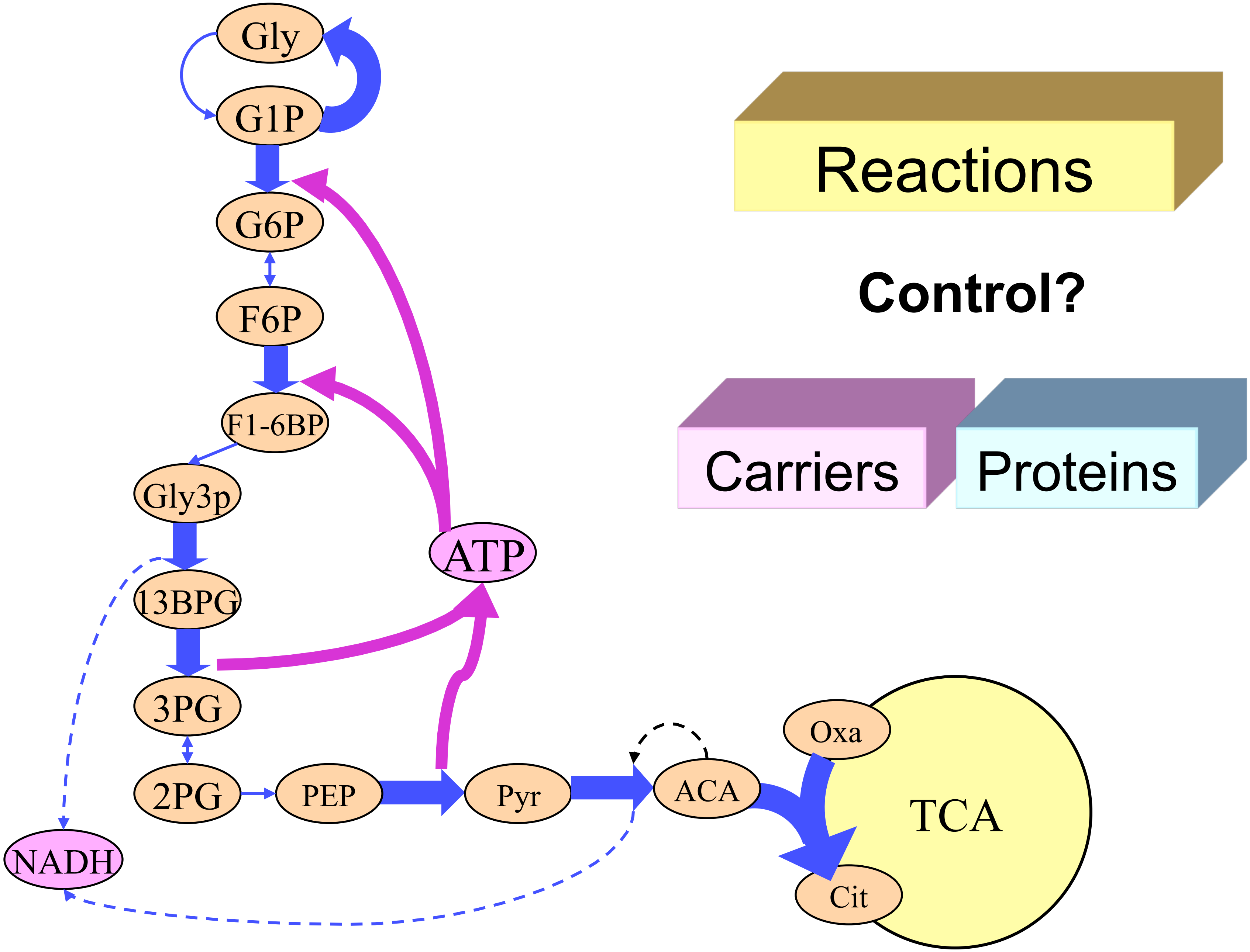
Enzymatically catalyzed reactions

Autocatalytic



Autocatalytic



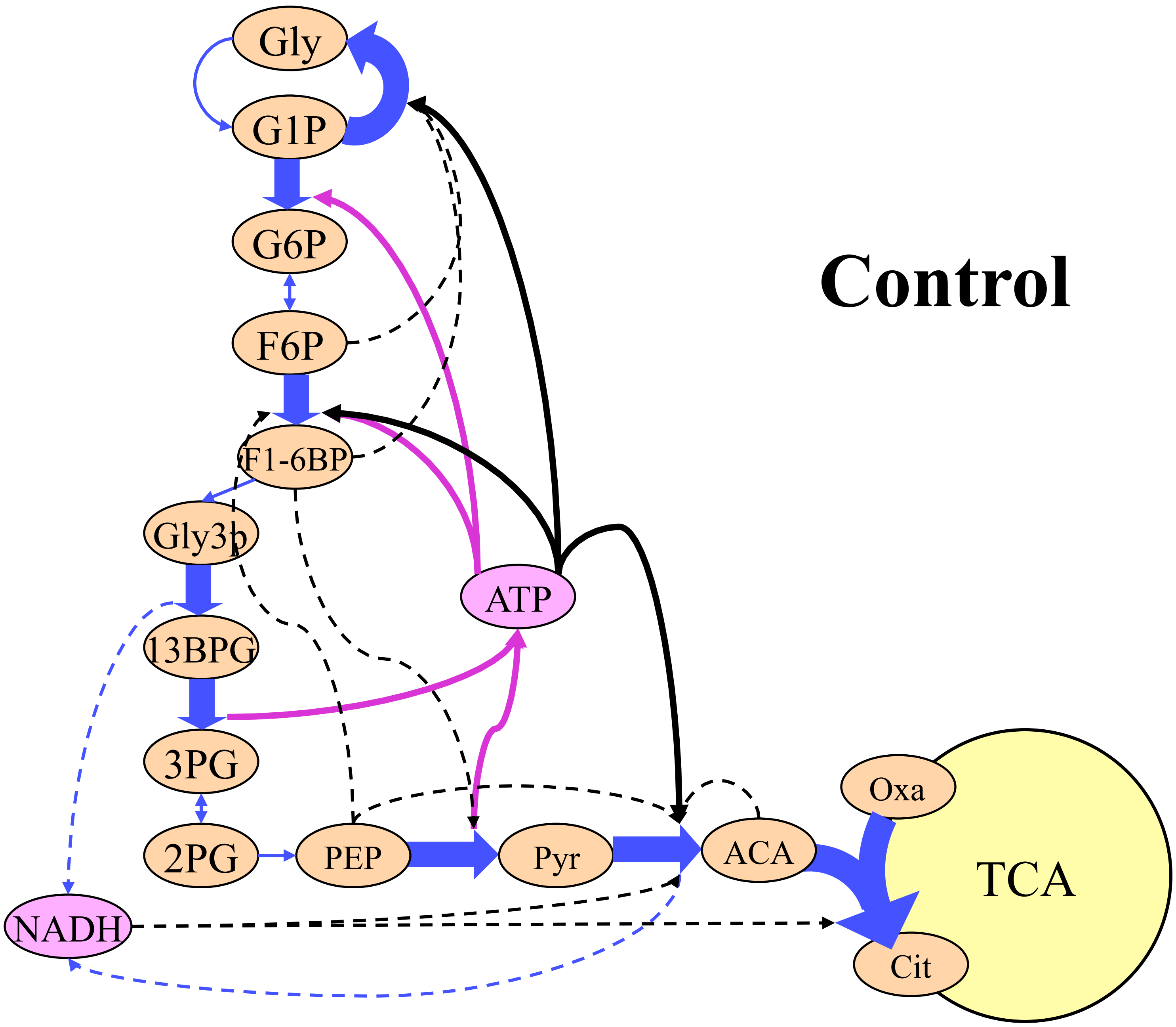


Reactions

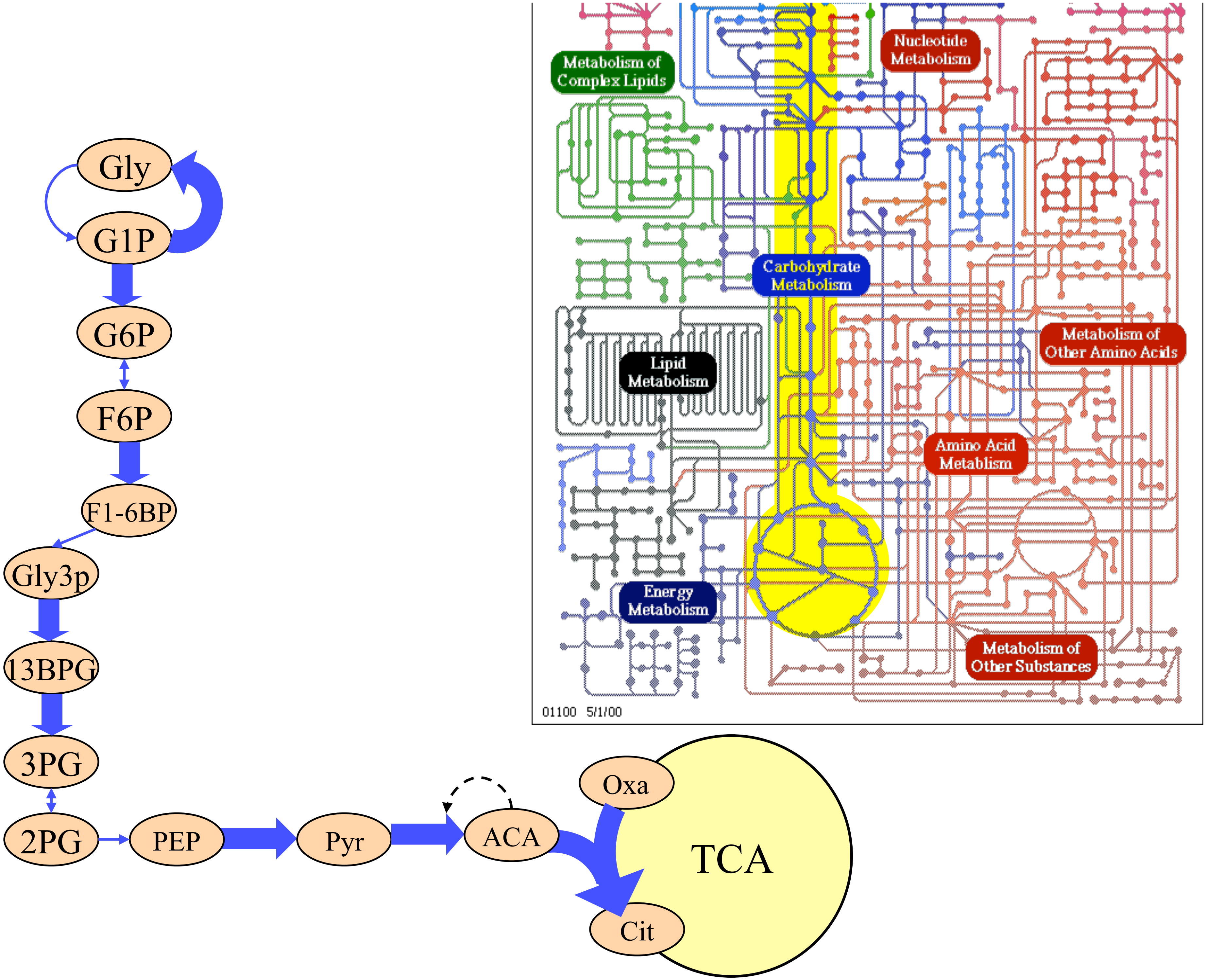
Control?

Carriers

Proteins


Gly
G1P
G6P
F6P
F1-6BP
Gly3p
13BPG
3PG
2PG
PEP
Pyr
ACA
ATP
NADH
Oxa
Cit
TCA

Control



Gly

G1P

G6P

F6P

F1-6BP

Gly3p

13BPG

3PG

2PG

PEP

Pyr

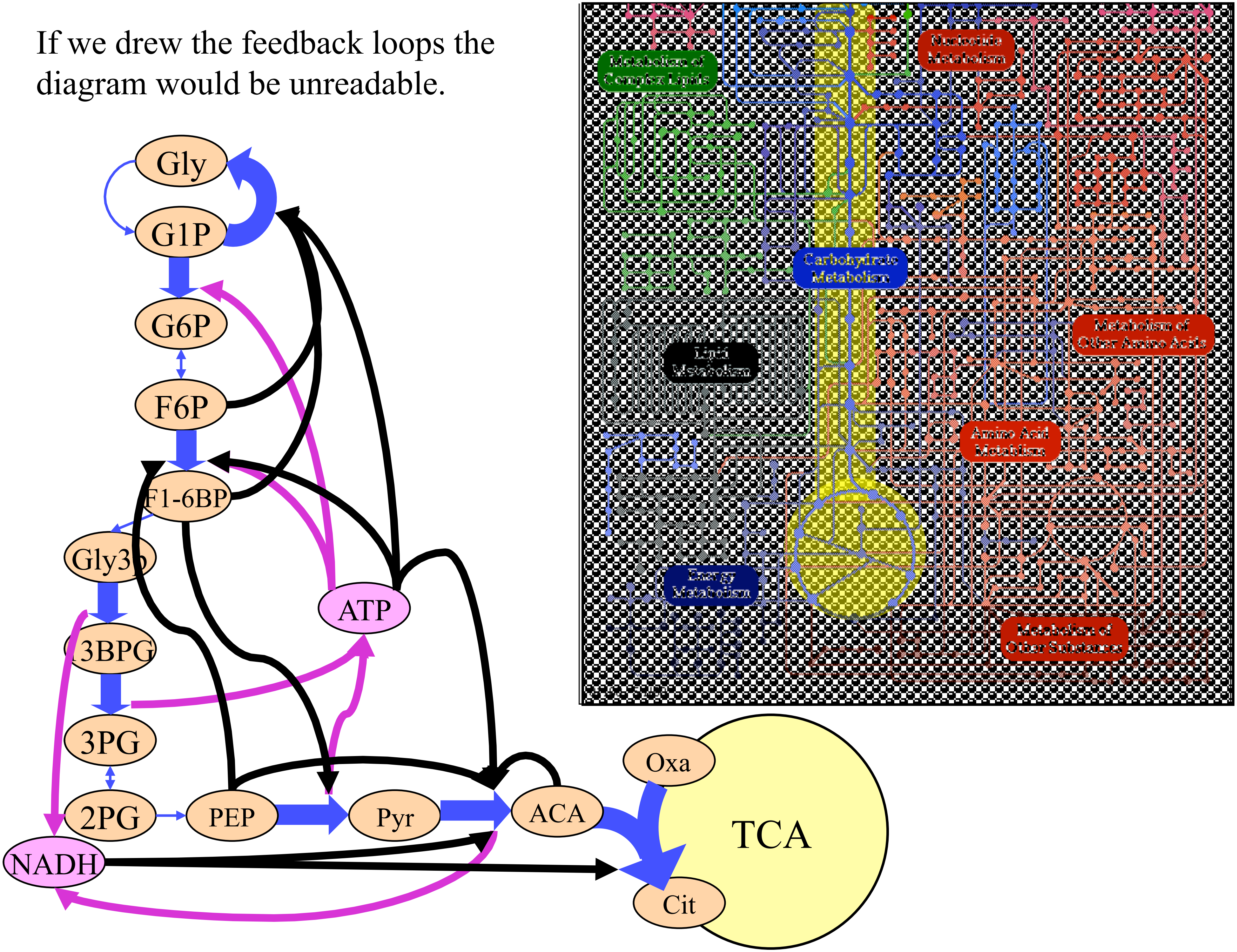
ACA

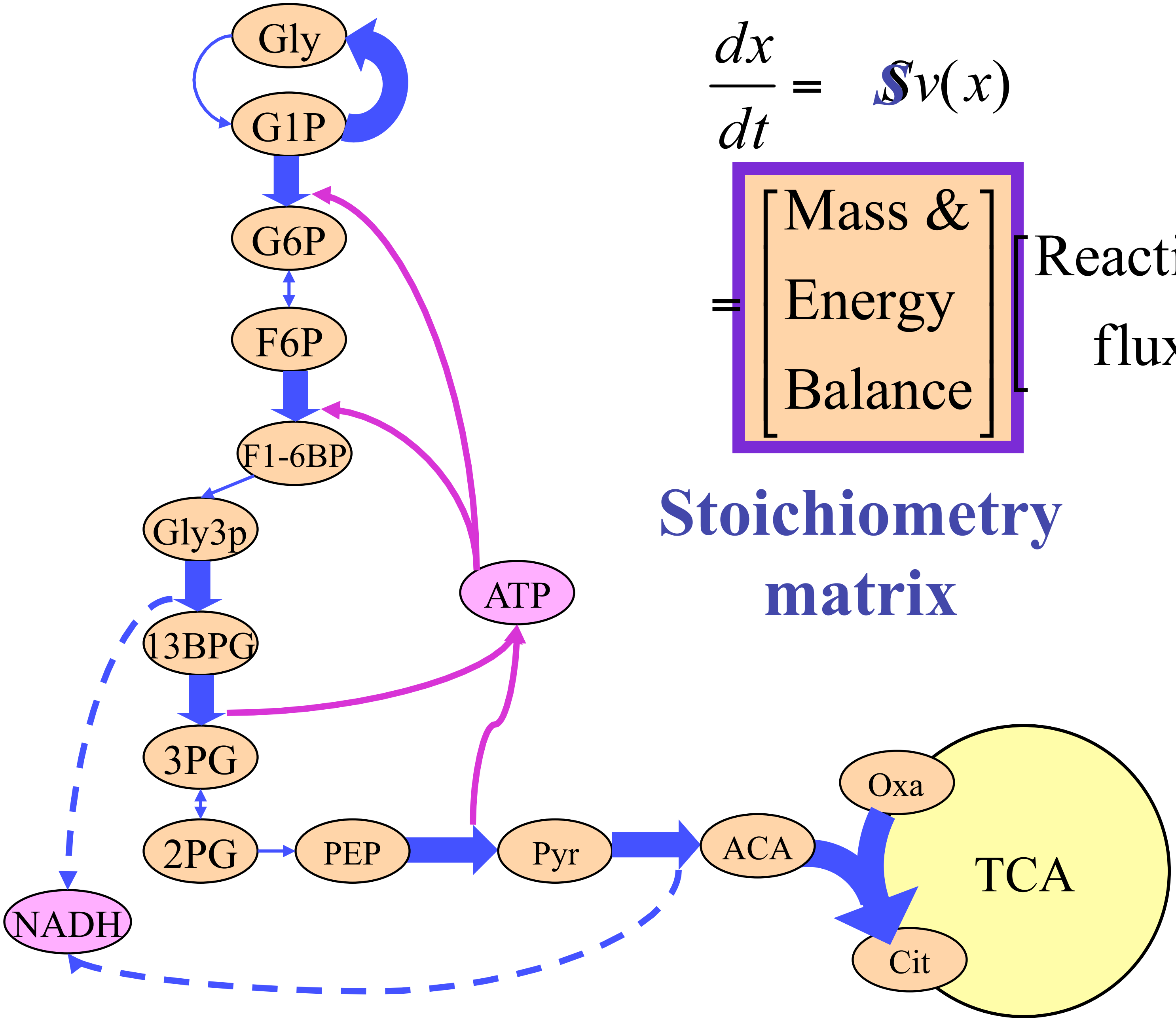
Oxa

TCA

Cit

If we drew the feedback loops the diagram would be unreadable.

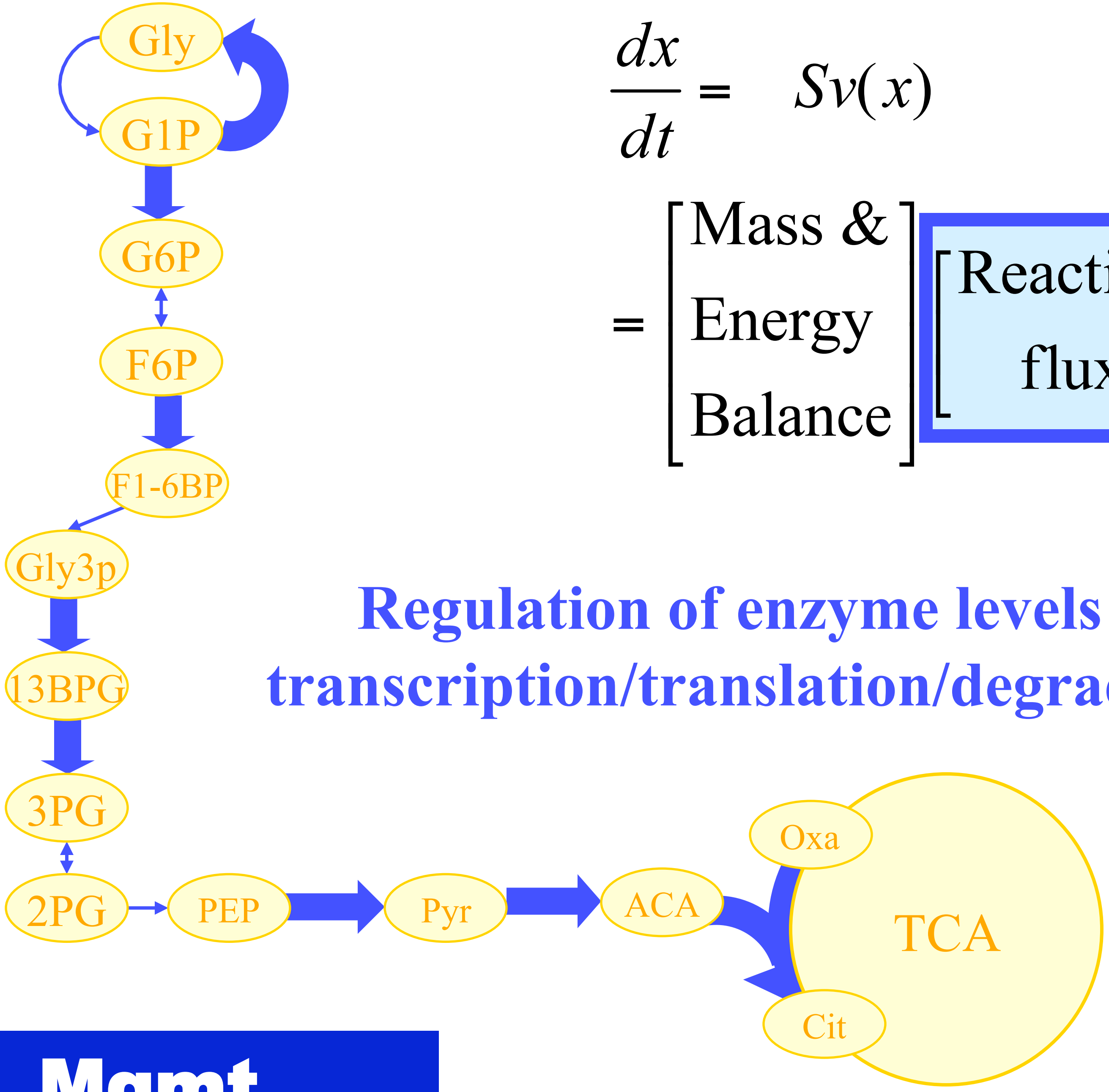




$$\frac{dx}{dt} = \mathbf{S}v(x)$$

$$= \boxed{\begin{bmatrix}\text{Mass \&}\\ \text{Energy}\\ \text{Balance}\end{bmatrix}} \begin{bmatrix}\text{Reaction}\\ \text{flux}\end{bmatrix}$$

Stoichiometry matrix

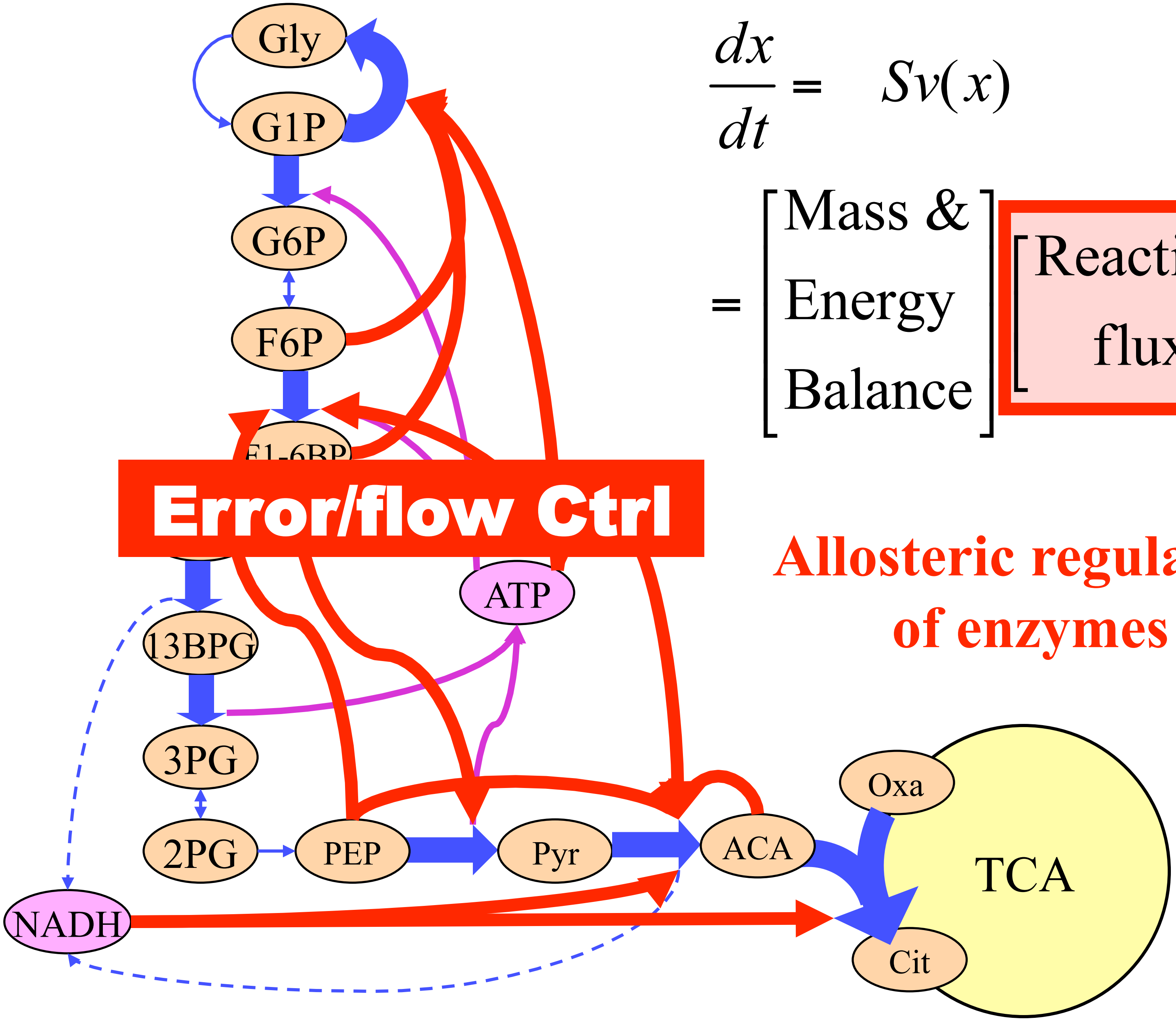


$$\frac{dx}{dt} = Sv(x)$$

$$= \begin{bmatrix}\text{Mass \&}\\ \text{Energy}\\ \text{Balance}\end{bmatrix}\boxed{\begin{bmatrix}\text{Reaction}\\ \text{flux}\end{bmatrix}}$$

Regulation of enzyme levels by transcription/translation/degradation

Mgmt



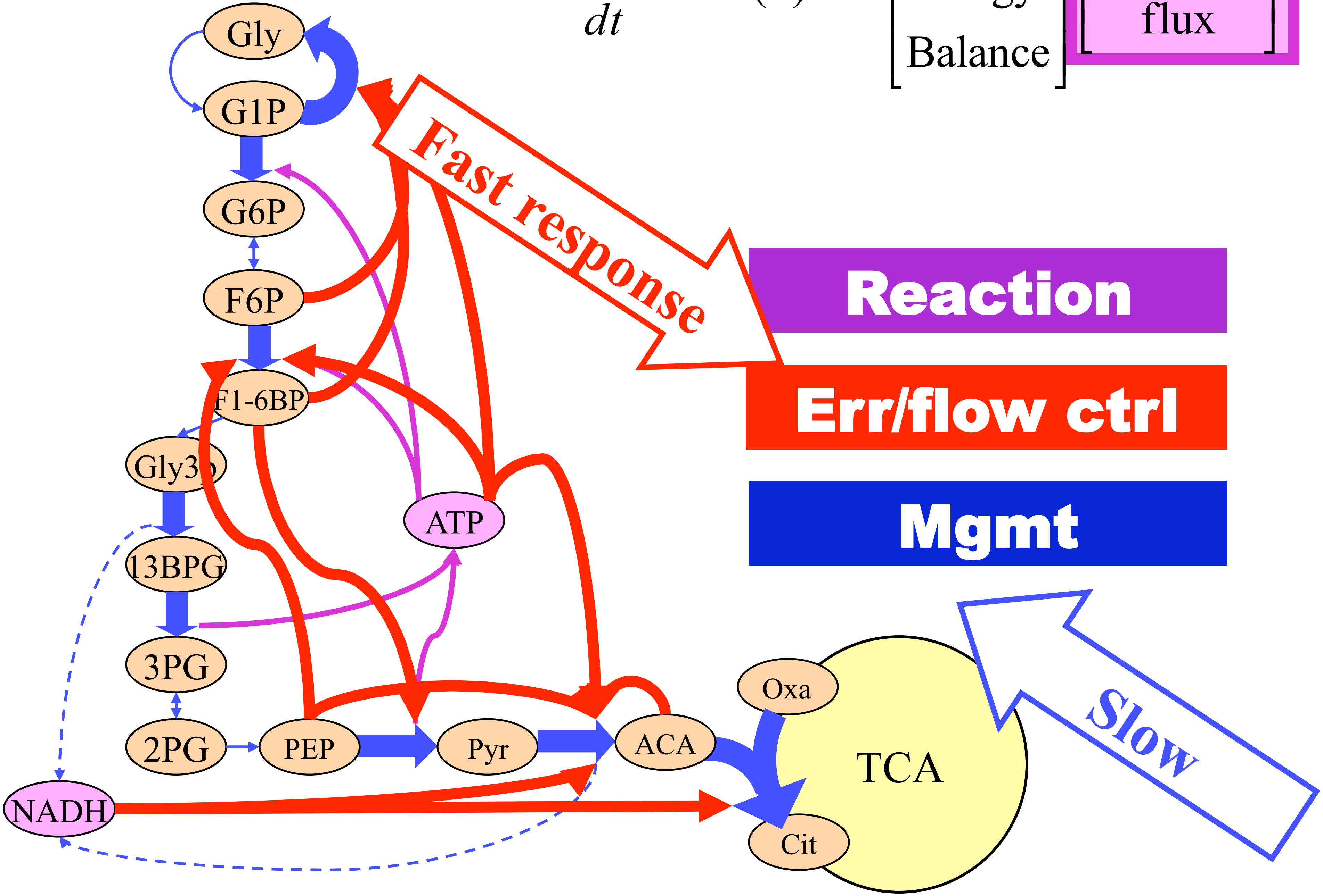
Error/flow Ctrl

$$\frac{dx}{dt} = Sv(x)$$

$$= \begin{bmatrix}\text{Mass \&}\\ \text{Energy}\\ \text{Balance}\end{bmatrix} \boxed{\begin{bmatrix}\text{Reaction}\\ \text{flux}\end{bmatrix}}$$

Allosteric regulation of enzymes

$$\frac{dx}{dt} = Sv(x) = \begin{bmatrix}\text{Mass \&}\\ \text{Energy}\\ \text{Balance}\end{bmatrix} \boxed{\begin{bmatrix}\text{Reaction}\\ \text{flux}\end{bmatrix}}$$



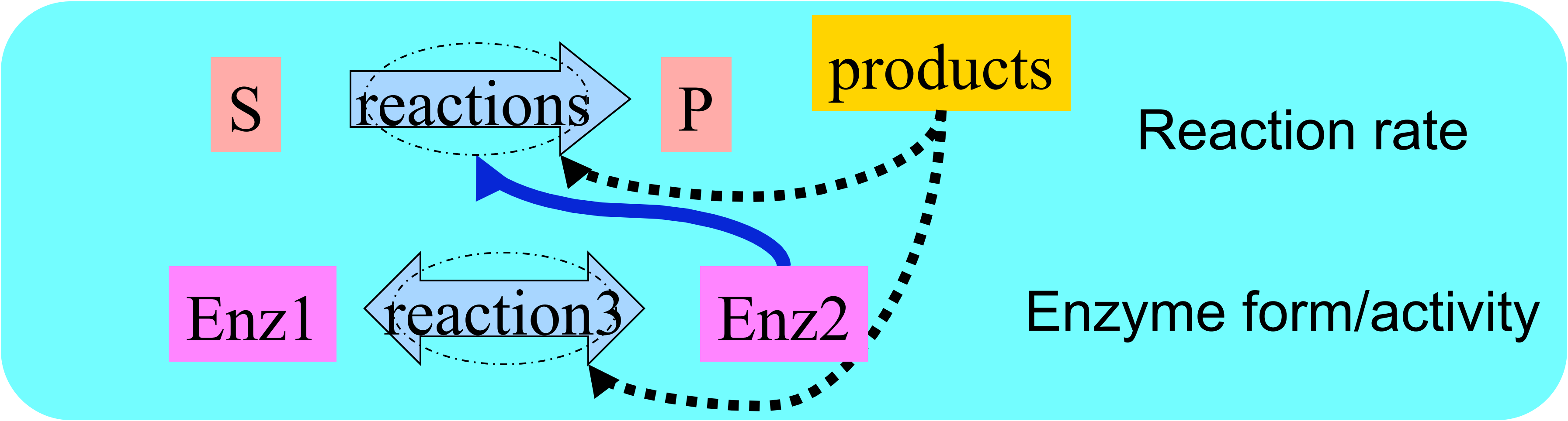
Fast response

Reaction

Err/flow ctrl

Mgmt

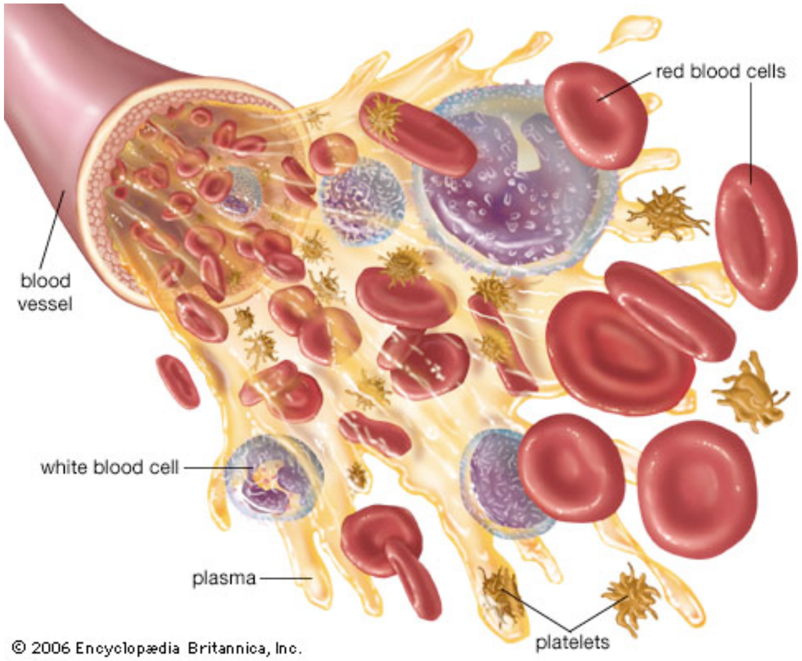
Slow



Reaction rate

Enzyme form/activity

Running only the top layers



© 2006 Encyclopædia Britannica, Inc.

Mature red blood cells live 120 days

“metabolism first” origins of life?

Reactions

Flow/error

Protein level

Reactions

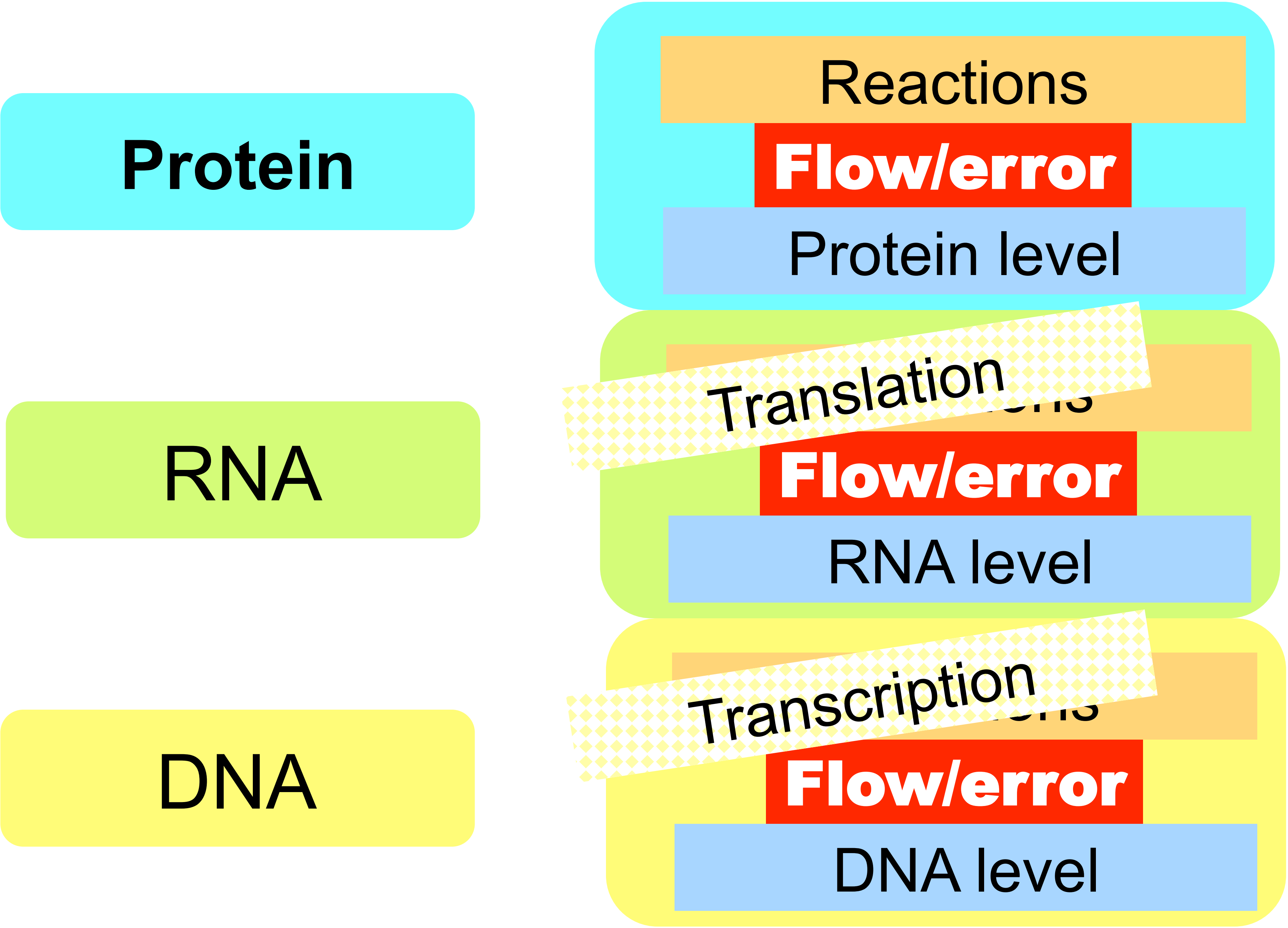
Flow/error

RNA level

Reactions

Flow/error

DNA level



Protein

RNA

DNA

Reactions

Flow/error

Protein level

Translation

ons

Flow/error

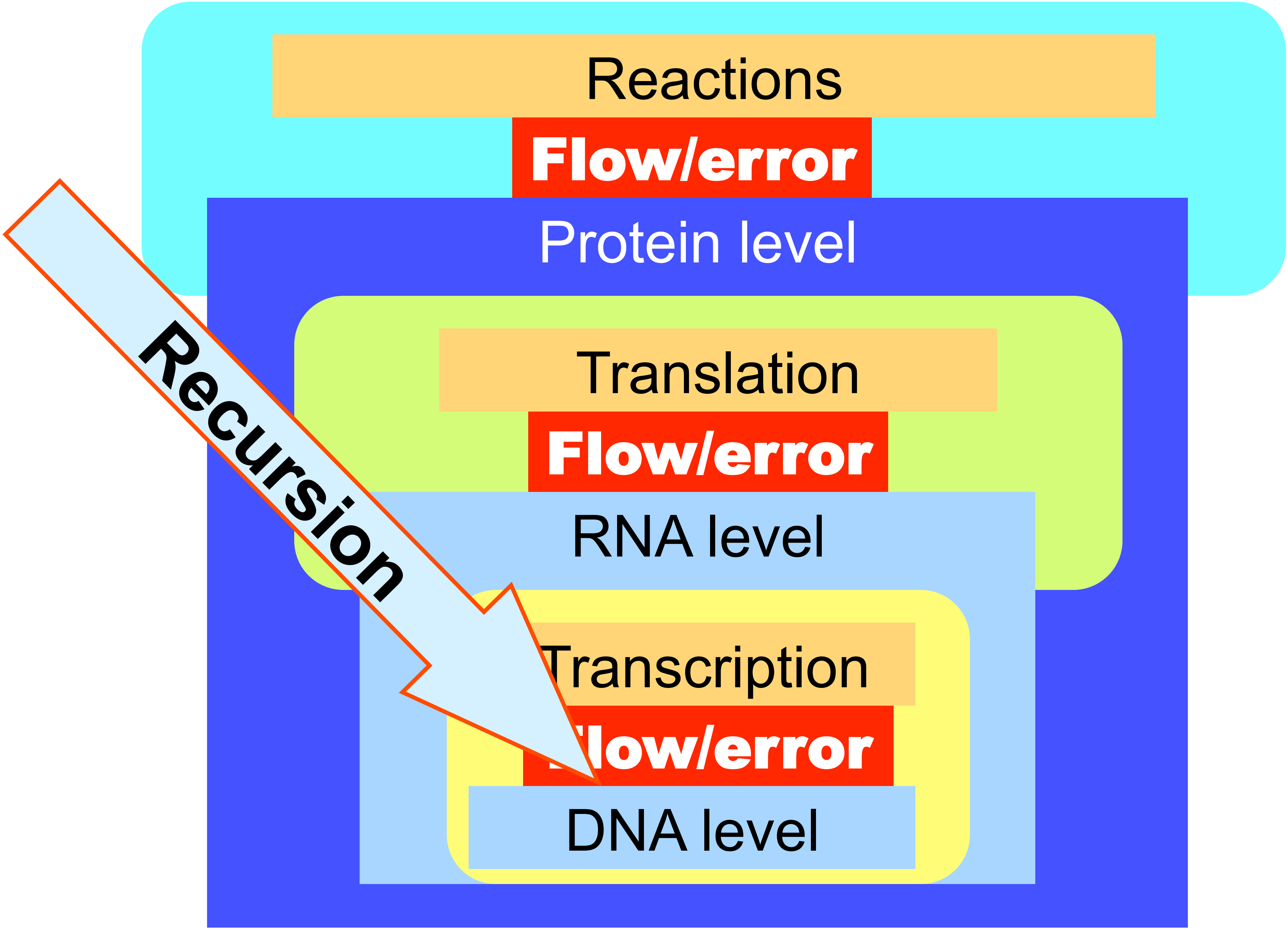
RNA level

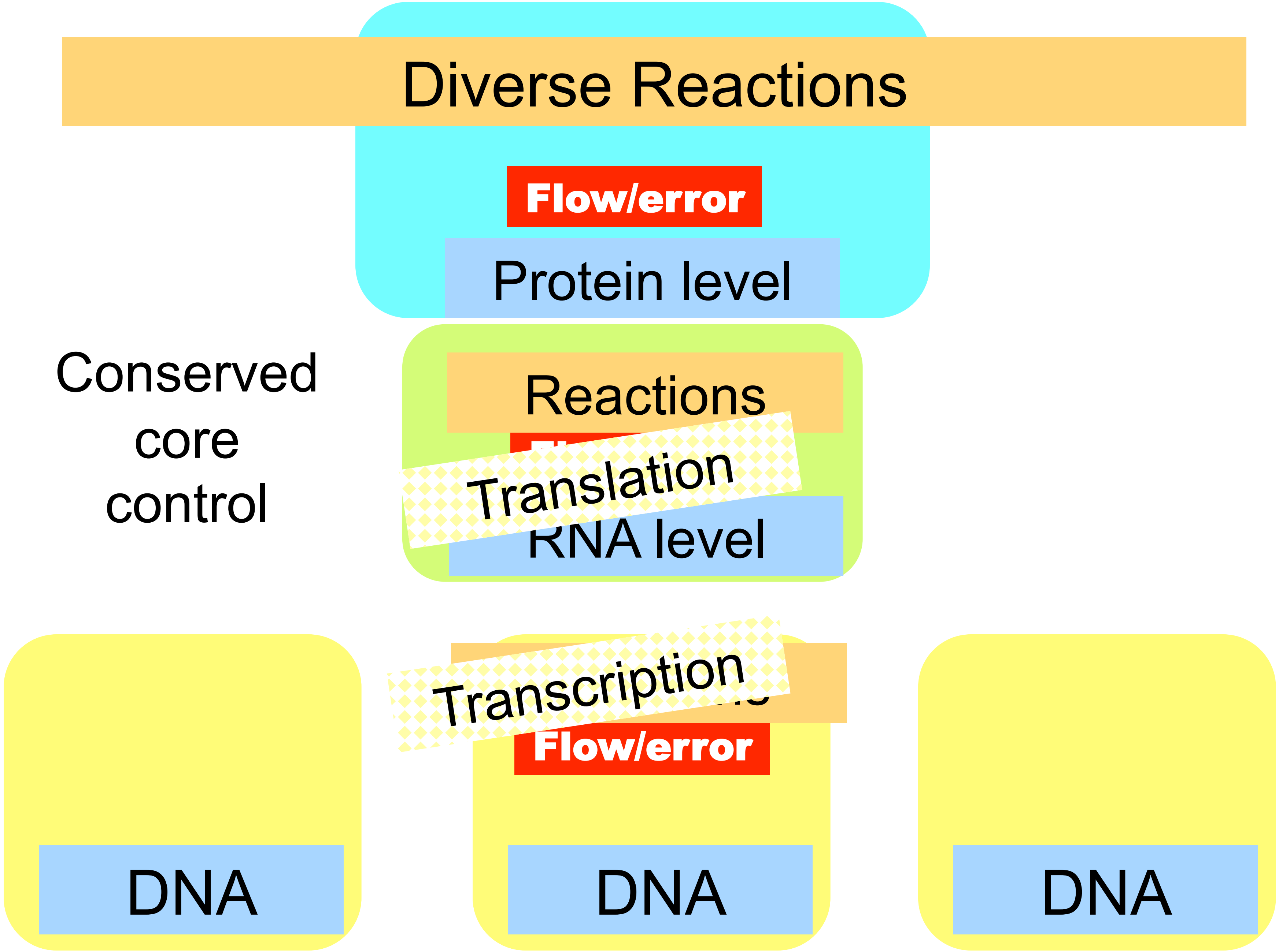
Transcription

ons

Flow/error

DNA level


Reactions
Flow/error
Protein level
Translation
Flow/error
RNA level
Transcription
Flow/error
DNA level
Recursion



Diverse Reactions

Flow/error

Protein level

Conserved core control

Reactions

Translation

RNA level

Transcription

Flow/error

DNA

DNA

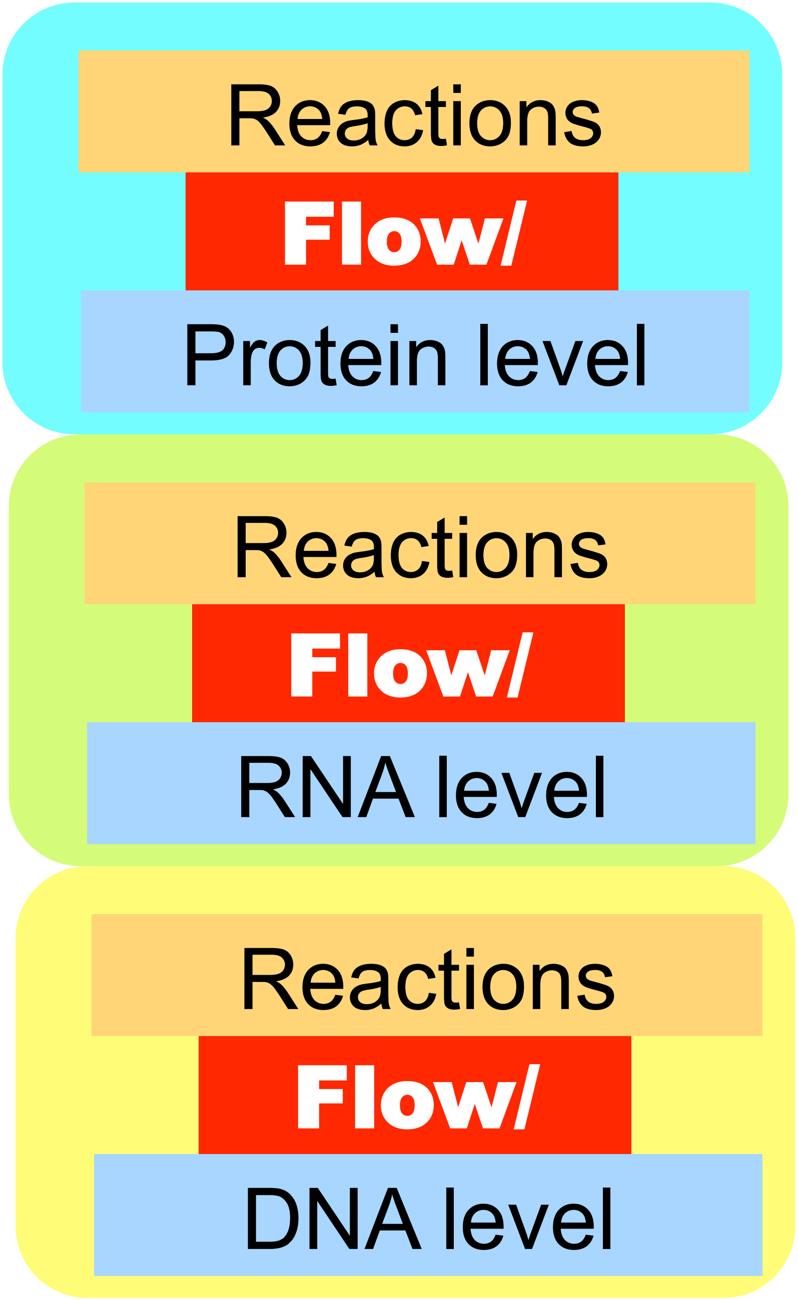
DNA

Diverse Genomes

Top to bottom

- Metabolically costly but fast to cheap but slow
- Special enzymes to general polymerases
- Allostery to regulated recruitment
- Analog to digital
- High molecule count to low (noise)

Rich Tradeoffs



Reactions

Flow/

Protein level

Reactions

Flow/

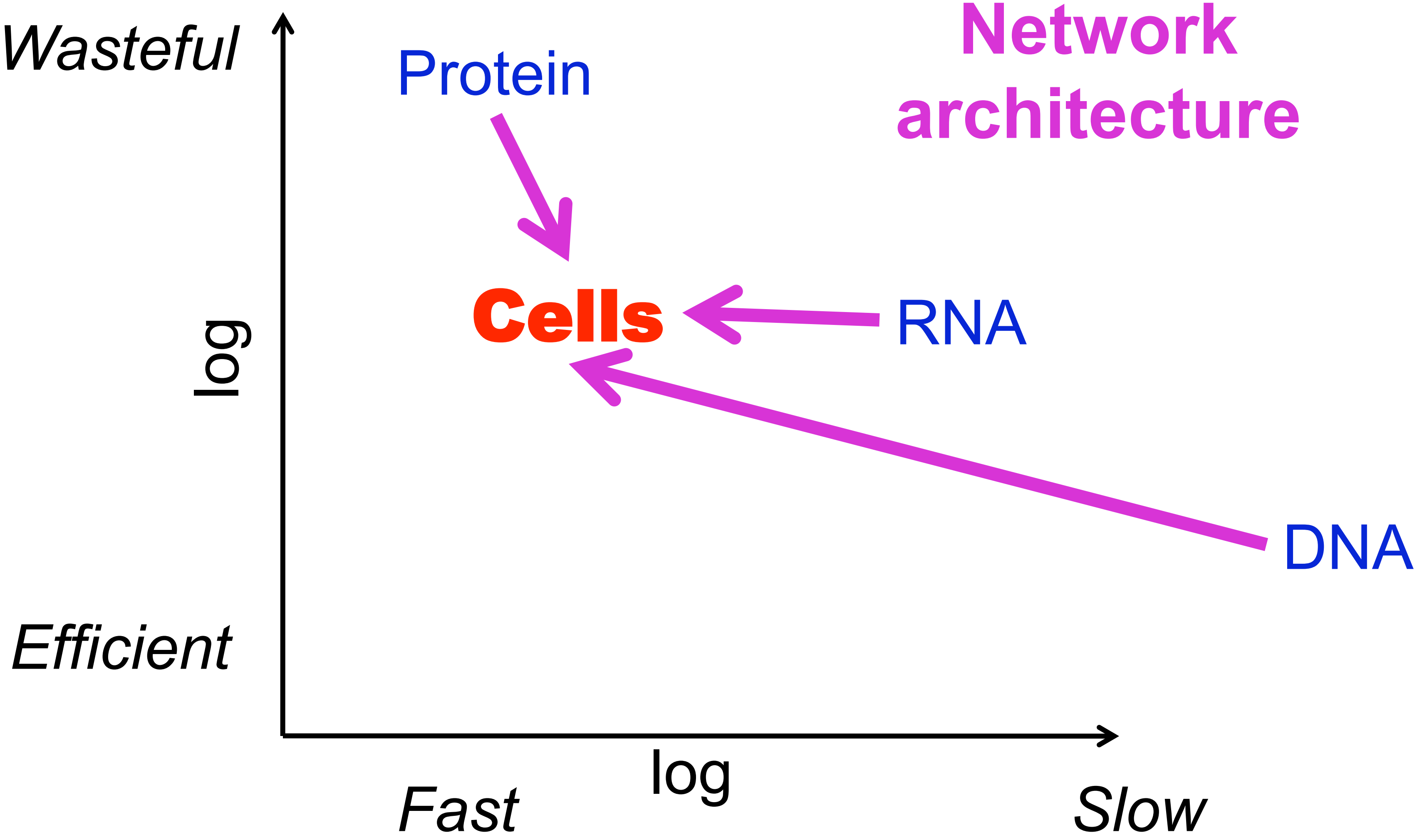
RNA level

Reactions

Flow/

DNA level

Network architecture



Wasteful

Efficient

log

log

Fast

Slow

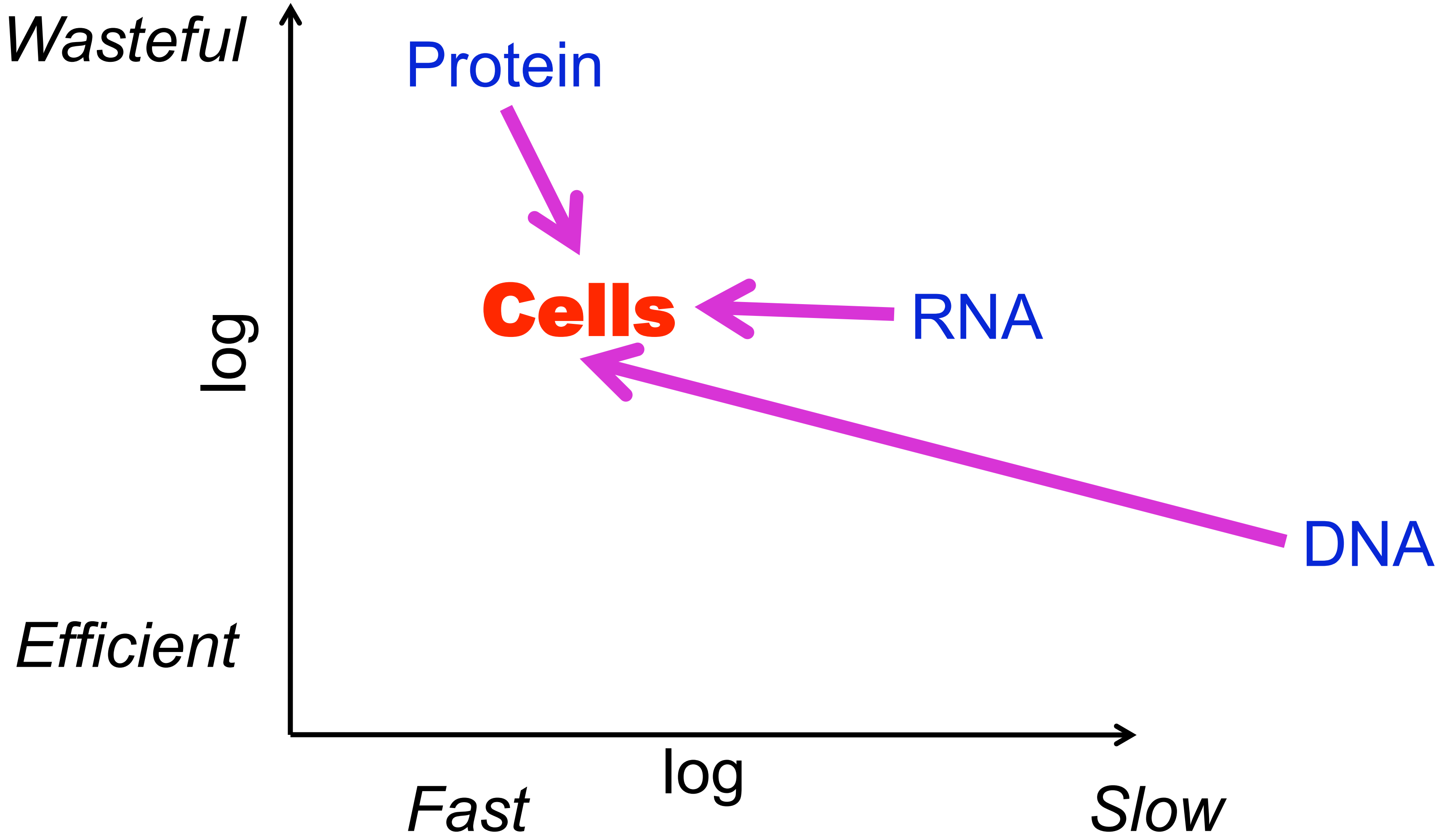
Protein

RNA

DNA

Cells

💣???☹



Wasteful

log

Efficient

Protein

Cells

RNA

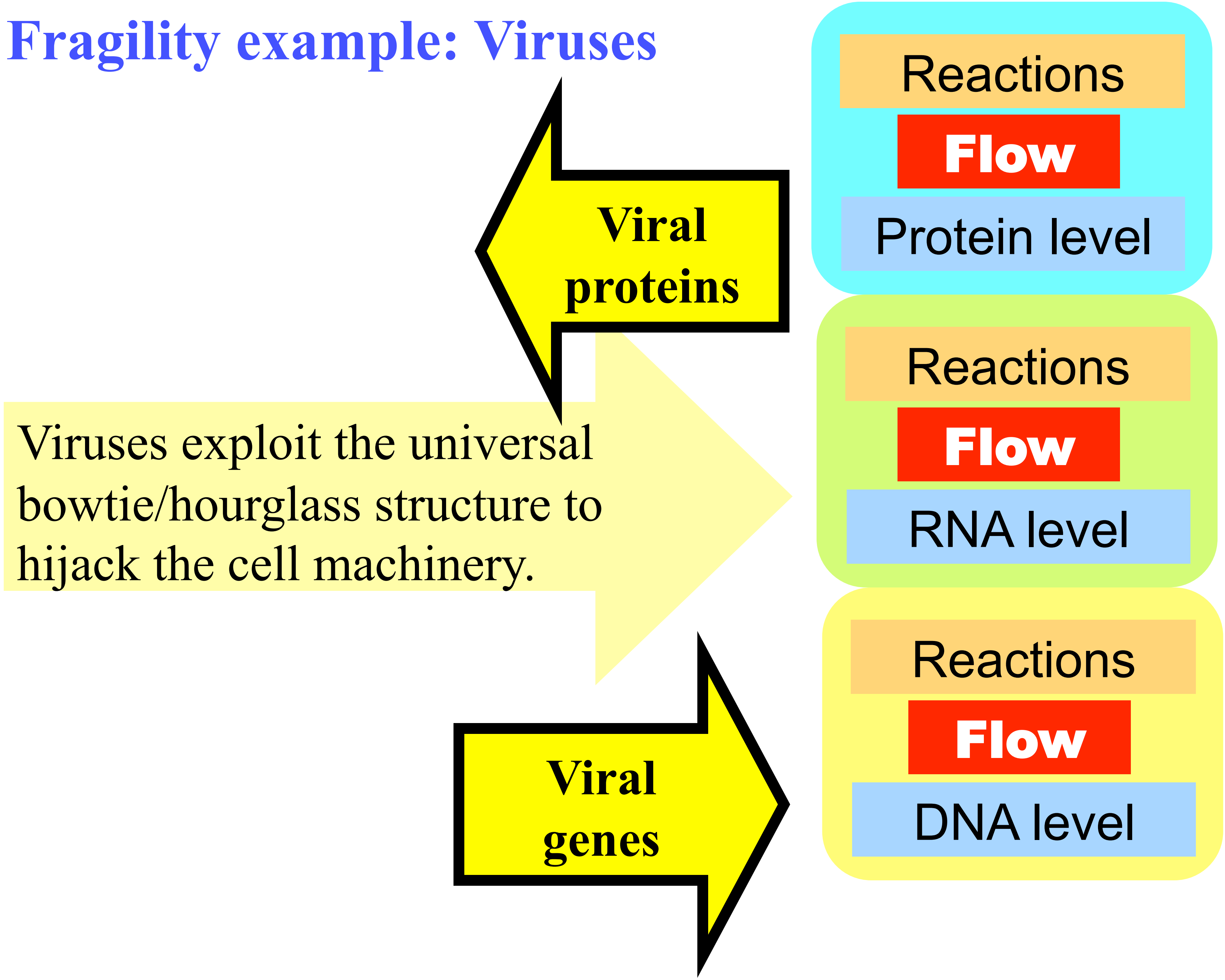
DNA

log

Fast

Slow

Fragility example: Viruses



Reactions
Flow
Protein level

Reactions
Flow
RNA level

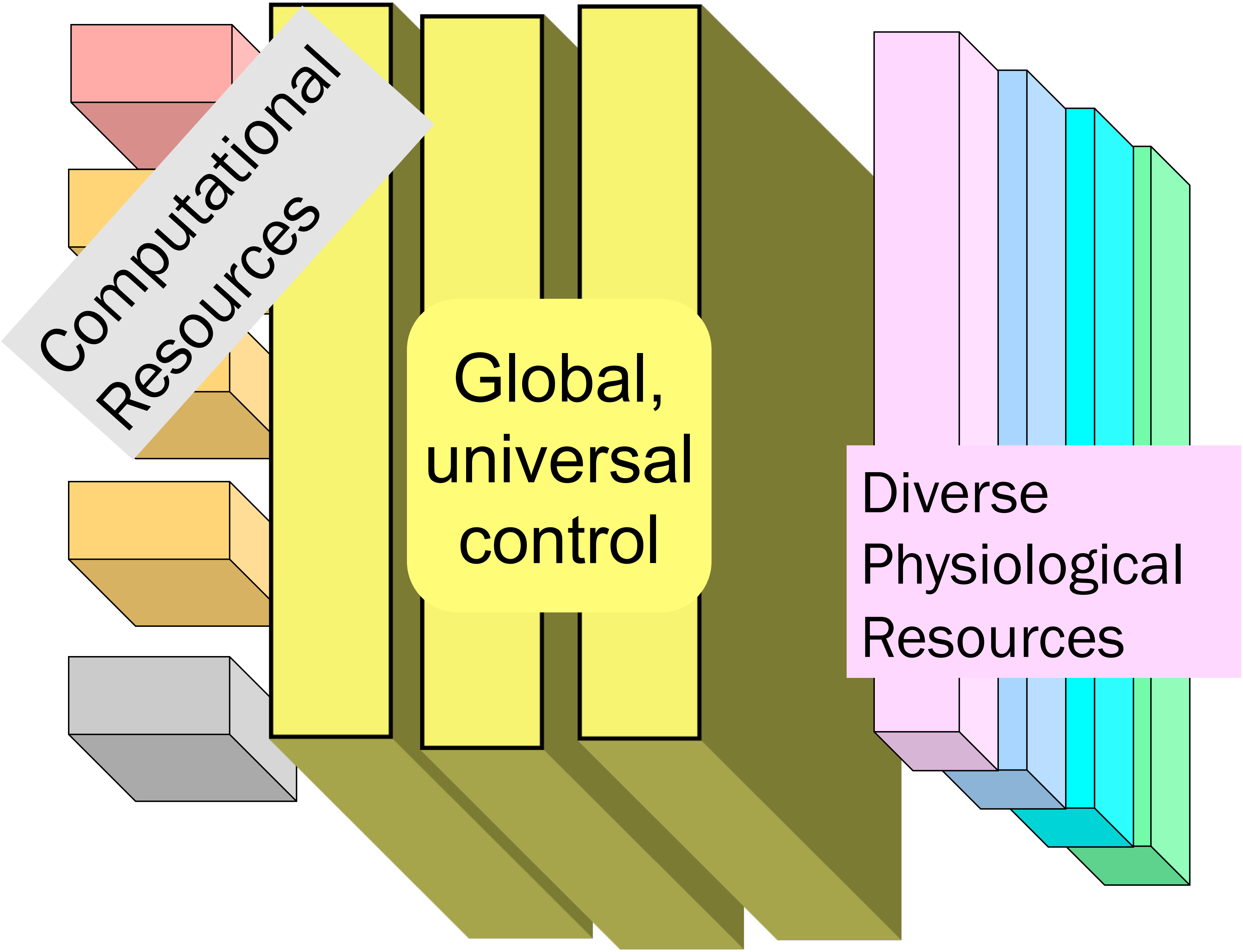
Reactions
Flow
DNA level

Viral proteins

Viruses exploit the universal bowtie/hourglass structure to hijack the cell machinery.

Viral genes

Layered Brain (Hawkins)?

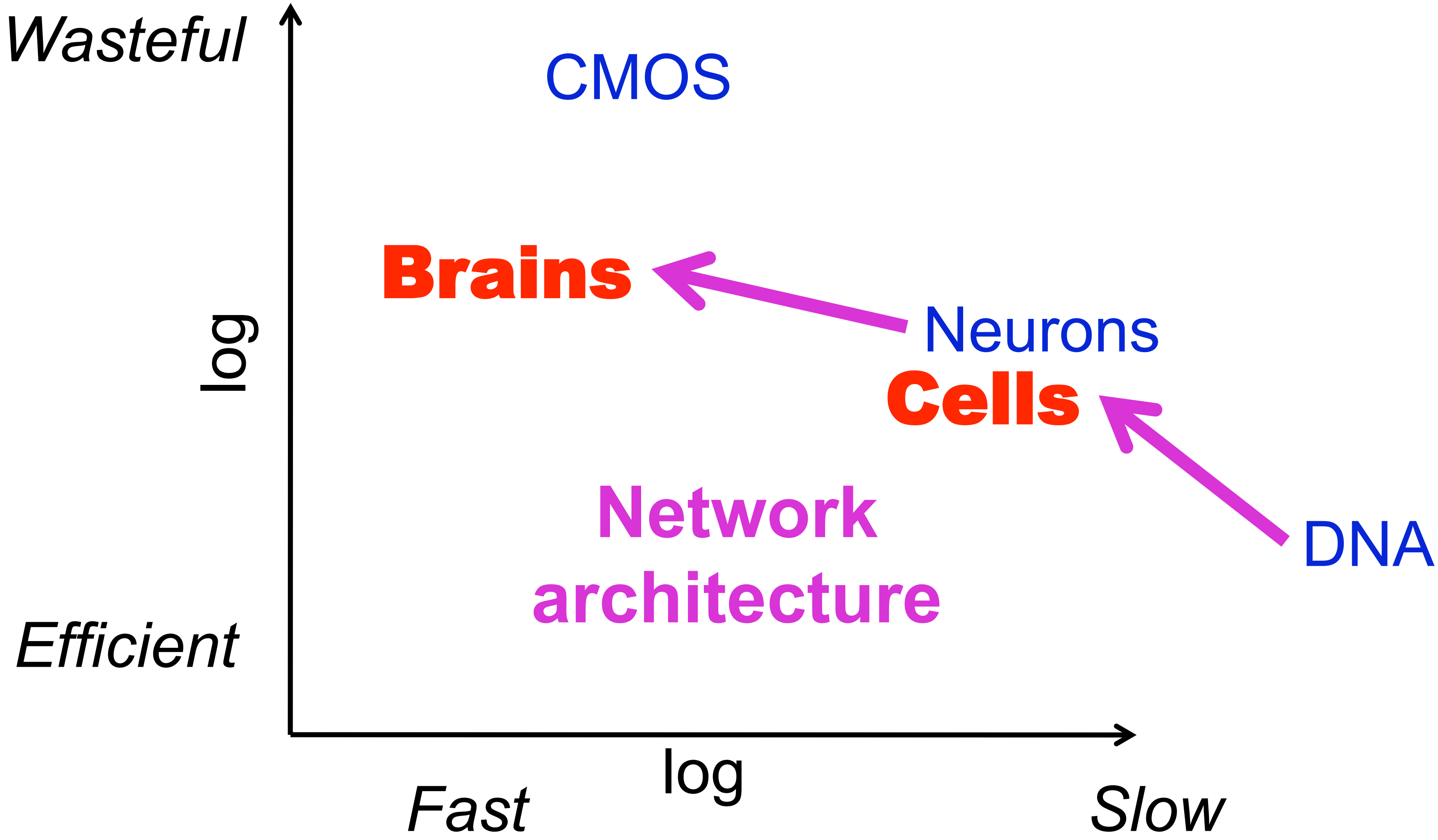


Computational Resources

Global, universal control

Diverse Physiological Resources

What makes this possible?



Wasteful

CMOS

Brains

Neurons

Cells

DNA

Network architecture

Efficient

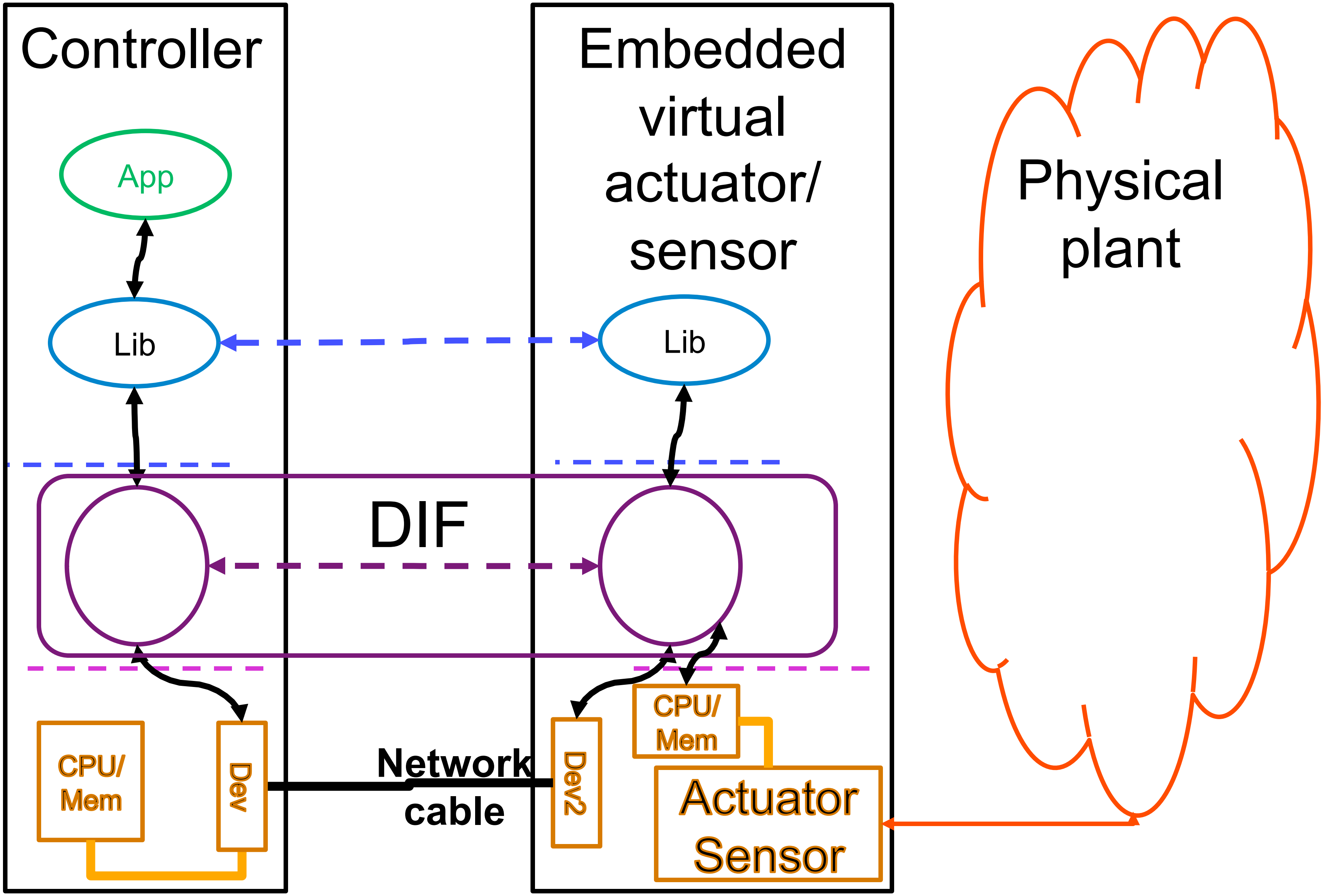
log

log

Fast

Slow

Networked embedded



Controller

App

Lib

Dev

CPU/ Mem

Network cable

DIF

Embedded virtual actuator/ sensor

Lib

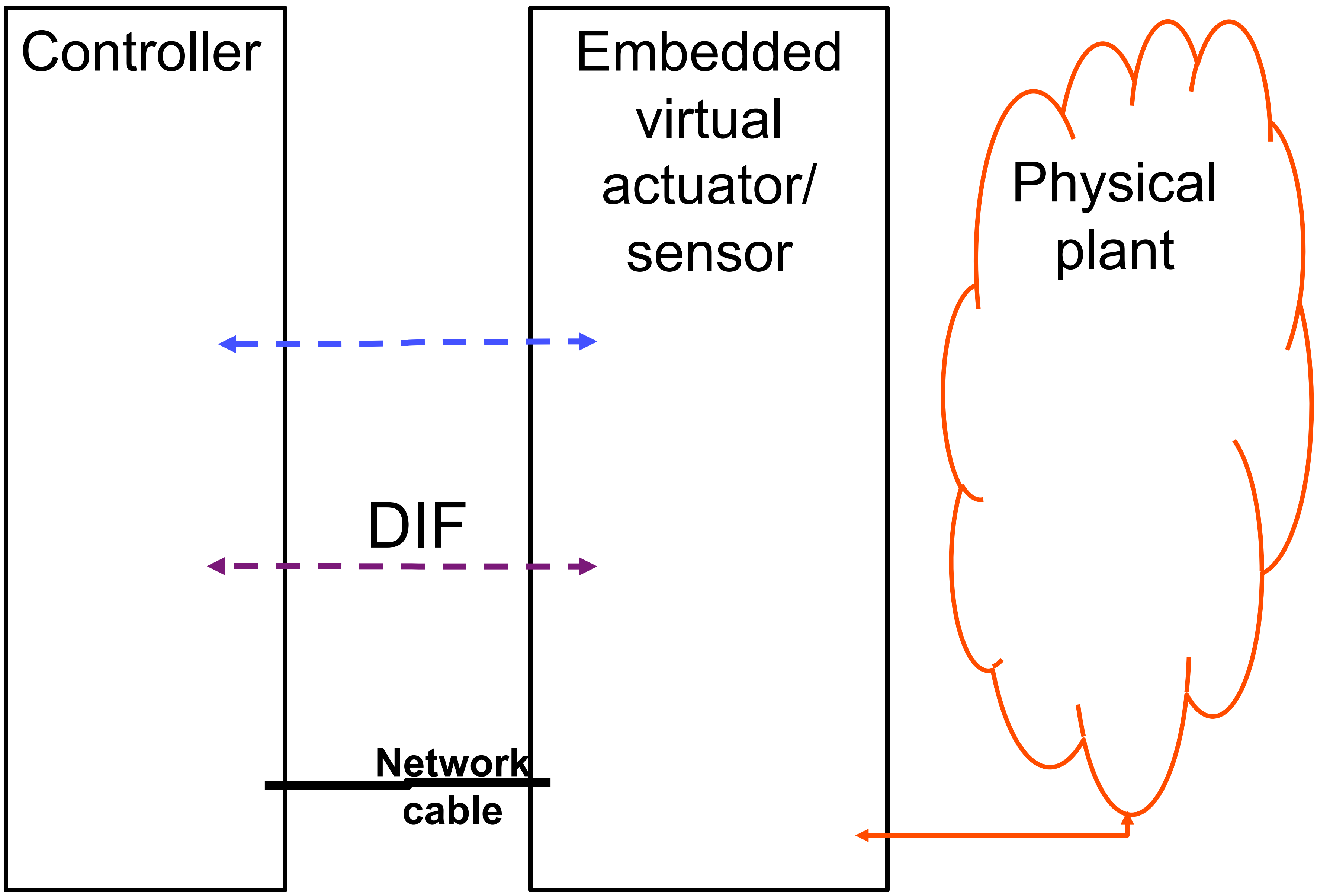
CPU/ Mem

Dev2

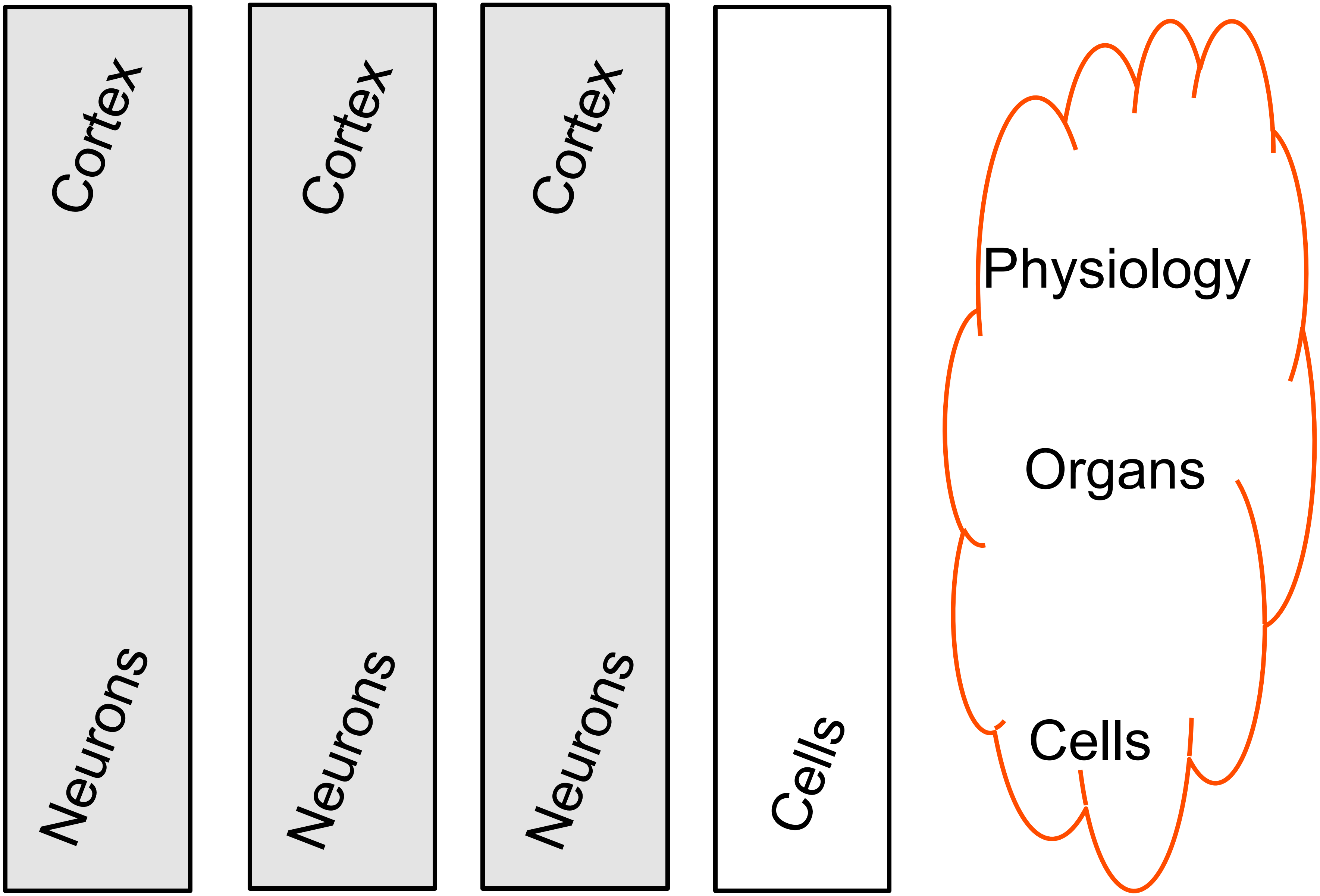
Actuator Sensor

Physical plant

Meta-layering of cyber-phys control



Meta-layers



Cortex

Neurons

Cortex

Neurons

Cortex

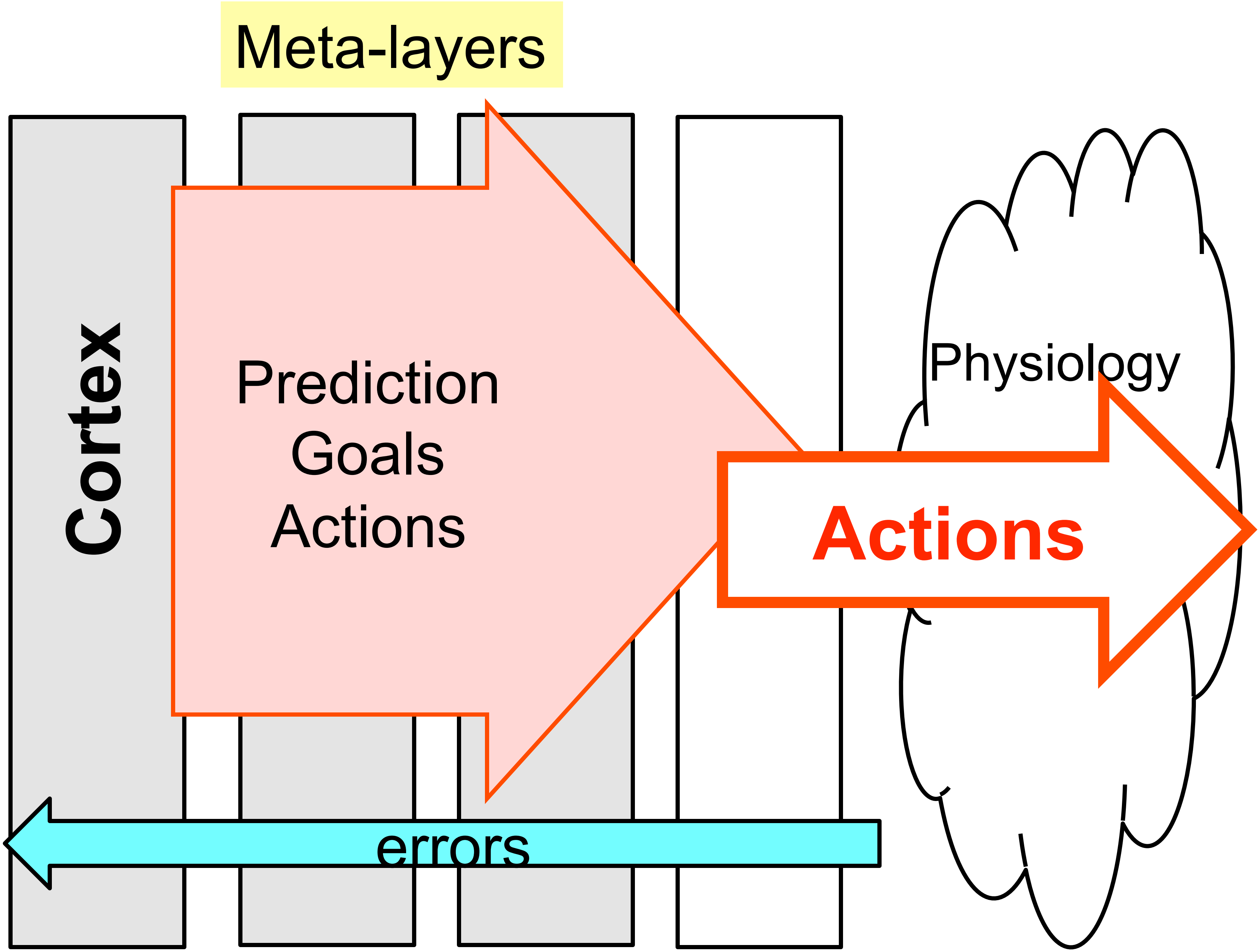
Neurons

Cells

Physiology

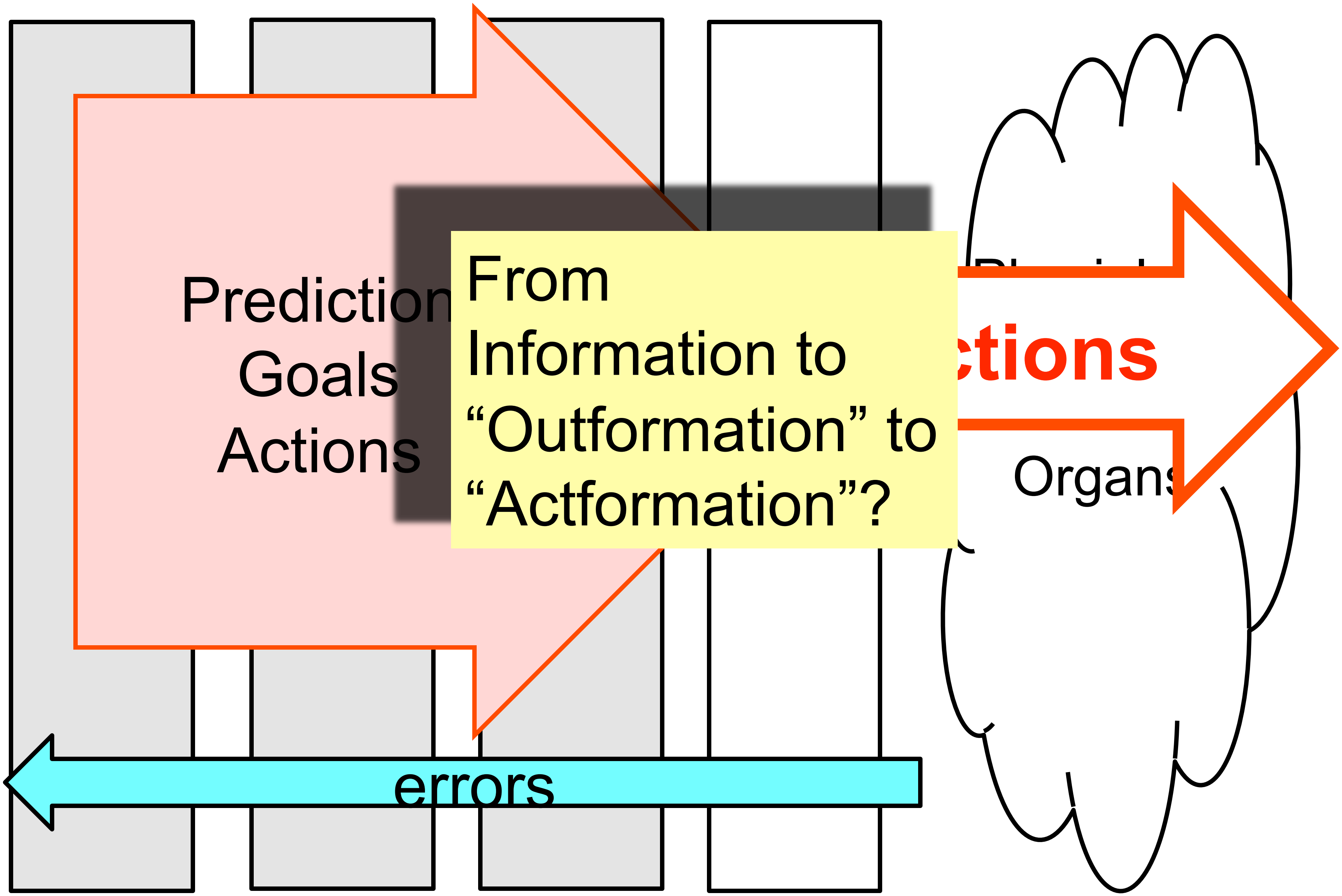
Organs

Cells



Meta-layers
Cortex
Prediction
Goals
Actions
Physiology
Actions
errors

Meta-layers



Prediction
Goals
Actions

ctions

Organs

errors

From
Information to
“Outformation” to
“Actformation”?

“Architecture” in practice

- Internet, biology, energy, manufacturing, transportation, water, food, waste, law, etc
- Many architectures are unsustainable/hard to fix

What does “architecture” mean here?

- Persistent, ubiquitous, global features
- Constrains the possible (for good or bad)
- Enables/prevents innovation, sustainability, etc,
- Theory is fragmented, incoherent, incomplete
- Needs rigor and relevance
- “Constraints that deconstrain” and “facilitated variation” (Gerhart and Kirschner)

Next steps?

- New course this term? (CDS 213?)
- Discuss at 1pm
- Flesh out details
- Integrated control, comms, computing, thermo/statmech, optimization, games, etc
- Motivated by very generic network challenges