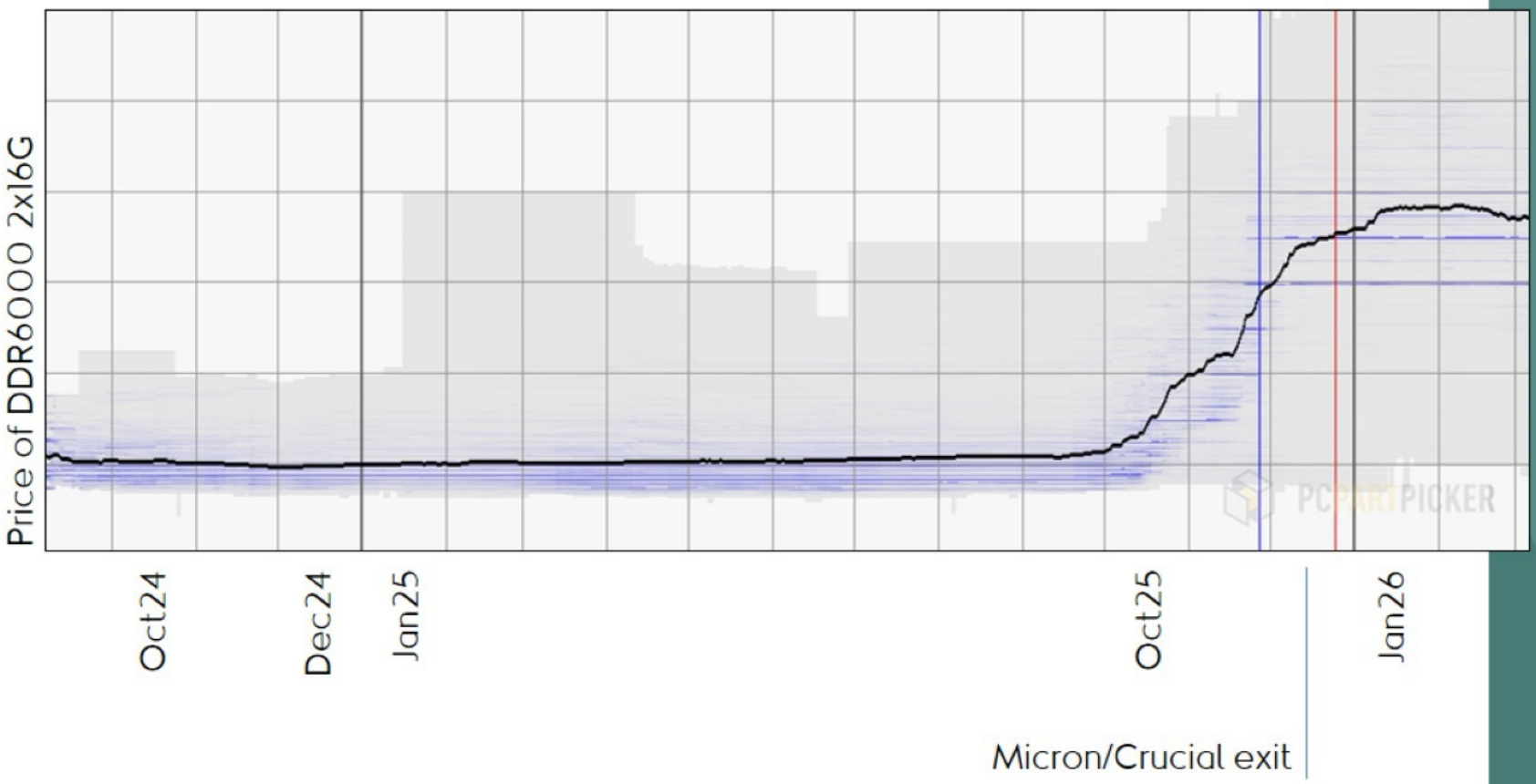
CONSTRAINED RESOURCE ARCHITECTURE

Russ White
Nokia

construction spend only
does not include servers or network

https://ourworldindata.org/grapher/monthly-spending-data-center-us

AI EXPLOSION



CHALLENGES: MEMORY

AI has pushed fiber demand to levels we've never seen ... BEAD is also finally rolling out at the same time. That dual demand elevates our fiber forecast, but it also makes the shortage more pronounced...
Clearfield

The U.S. will need to add 213.3 million more fiber miles by 2029, more than doubling its current amount from 159.6 million fiber miles to 372.9 million miles
Corning

We've seen the writing on the wall for awhile that data centers need fiber and lots of it. Research from *RVA LLC* has now done the math and worked out that providers need to build about 92,000 new route miles in the next five years to support that demand.

CHALLENGES: OPTICS

CHALLENGES: ROUTERS

Supply chain constraints and long lead times for components, ranging from 38 to 52 weeks, are impacting (X's) ability to meet demand.

https://www.gurufocus.com/

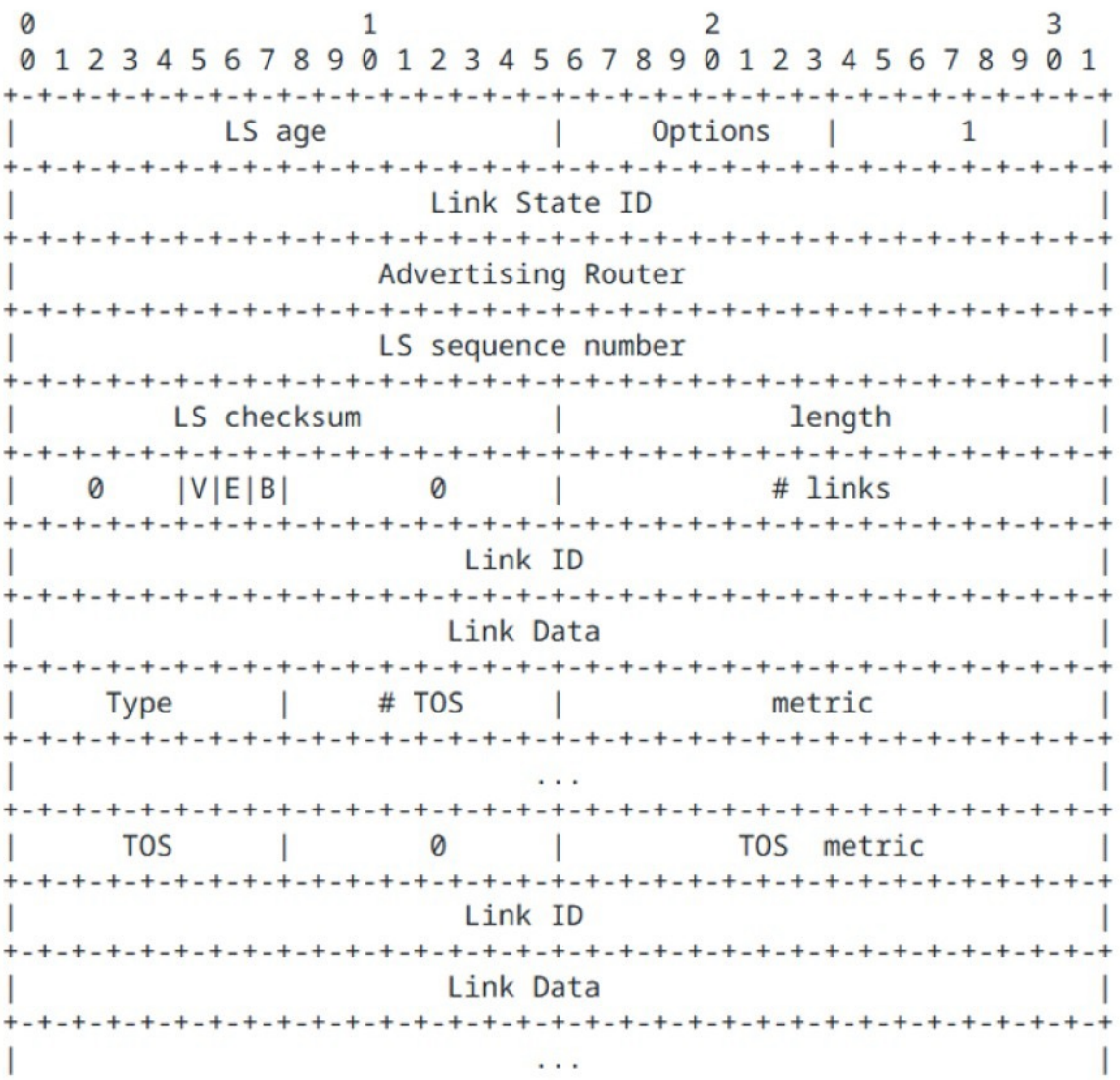
The worldwide supply chain crisis, beginning with closures in 2020, spiraled into a multi-year problem for the hardware industry. Due to severe chip and other component shortages, lead times turned from weeks to months.

https://info.pivitglobal.com/blog/update-on-lead-times

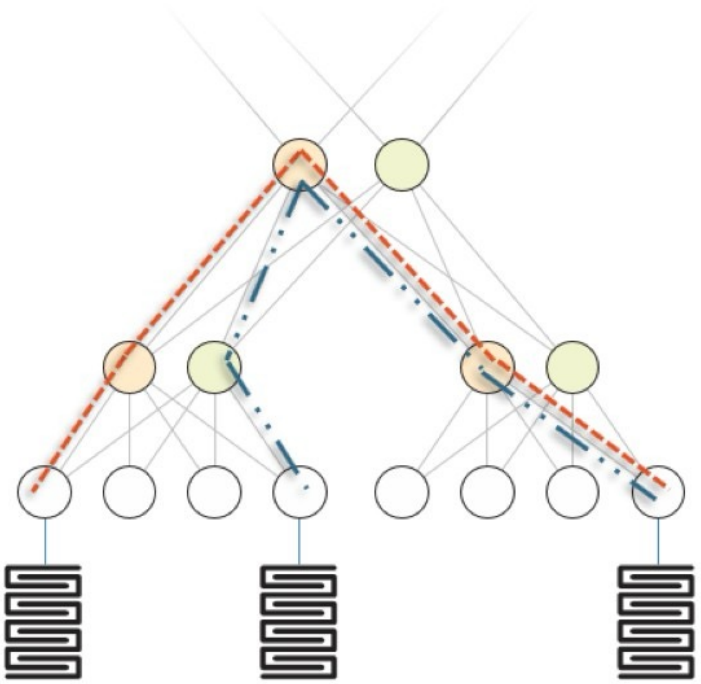
Problems seem to have eased up somewhat in some areas …

…but not in others

…we should still think about lessons learned, how we can design better, etc.



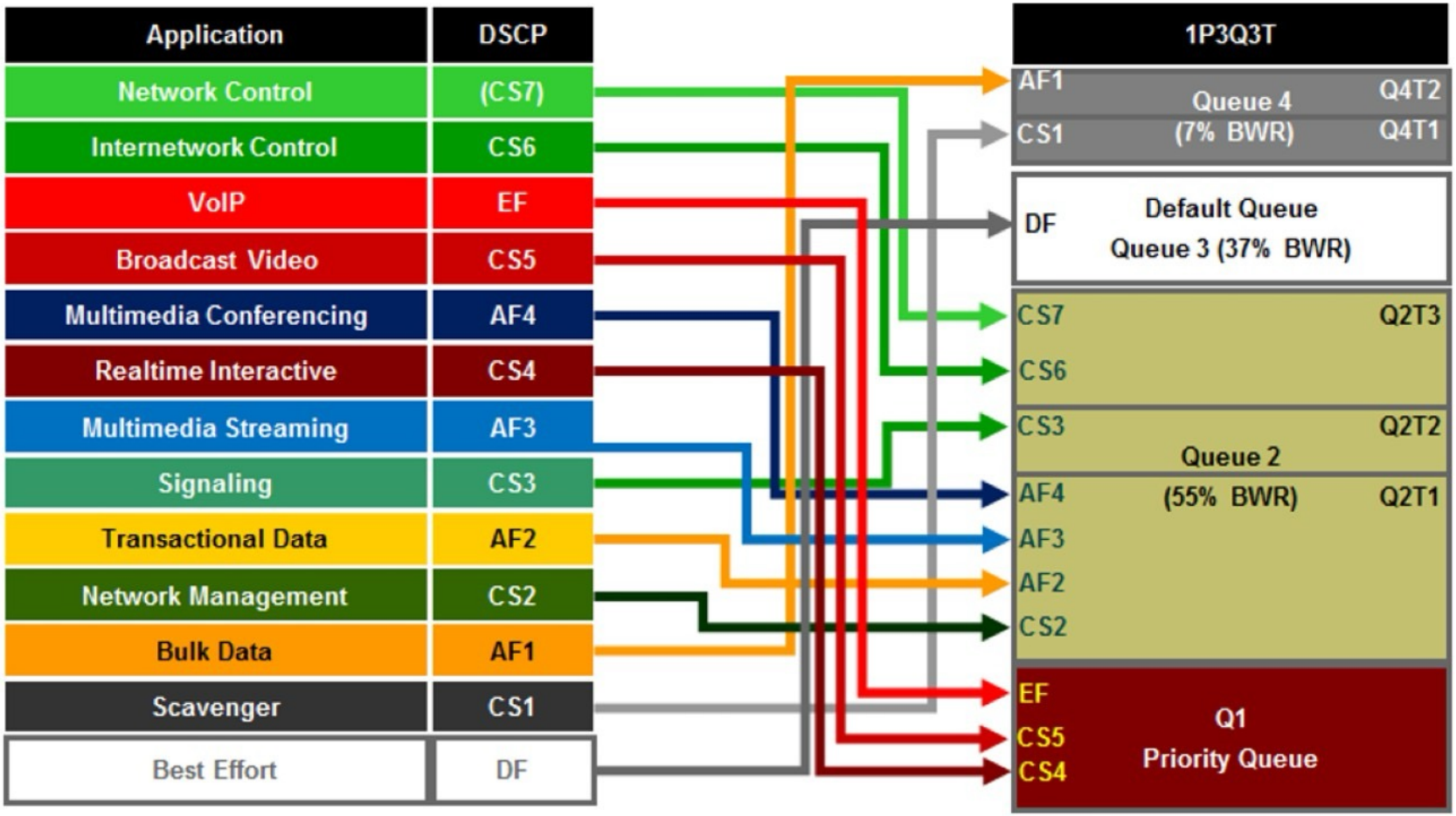
ENCODING EFFICIENCY



low ECMP fanout means sharing paths much earlier in the network

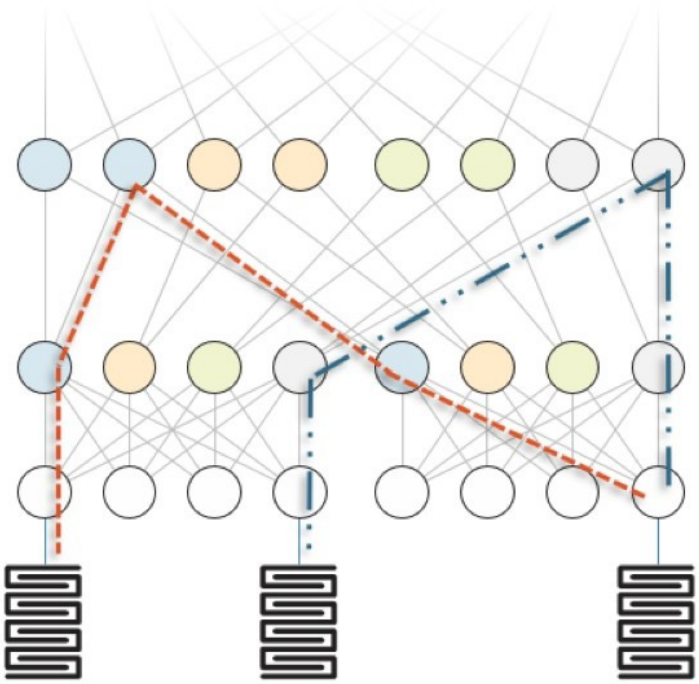
"bulk up the links!"

SCALE UP



https://easyqos-16.readthedocs.io/en/latest/chapter-09.html

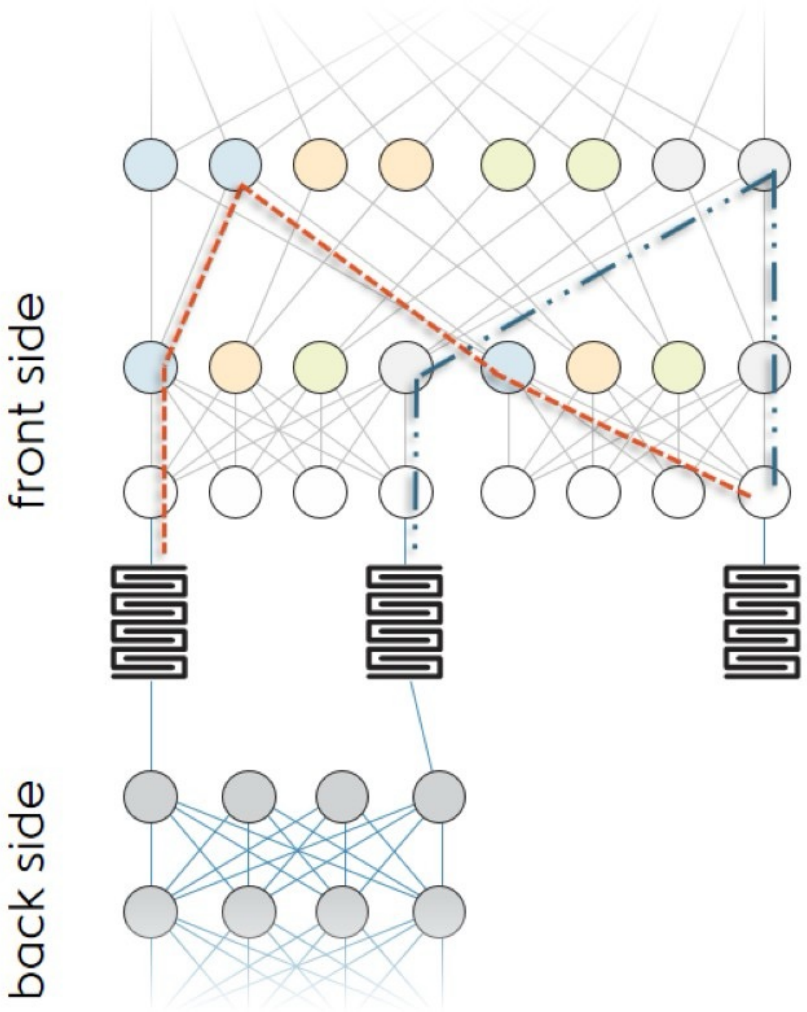
QUALITY OF SERVICE



build out, not up

opens opportunities for new ways to support quality of service

SCALE OUT



load placement

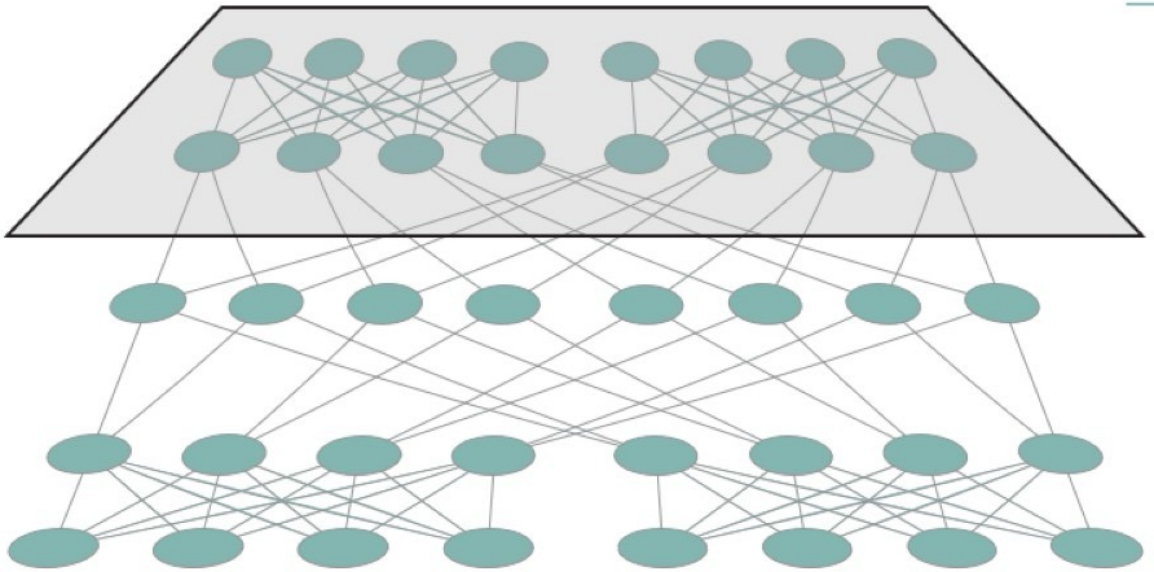
traffic placement

scheduling

TRAFFIC STEERING

Simplified?	Complexified?	Lived with?
Queueing	Parallelism	Protocol encoding complexity
Circuit sizing	Traffic placement	
	Traffic scheduling	
	Load placement	
Fewer big things *of many kinds*	More smaller things *of fewer kinds*	

How do we manage more small things more effectively?

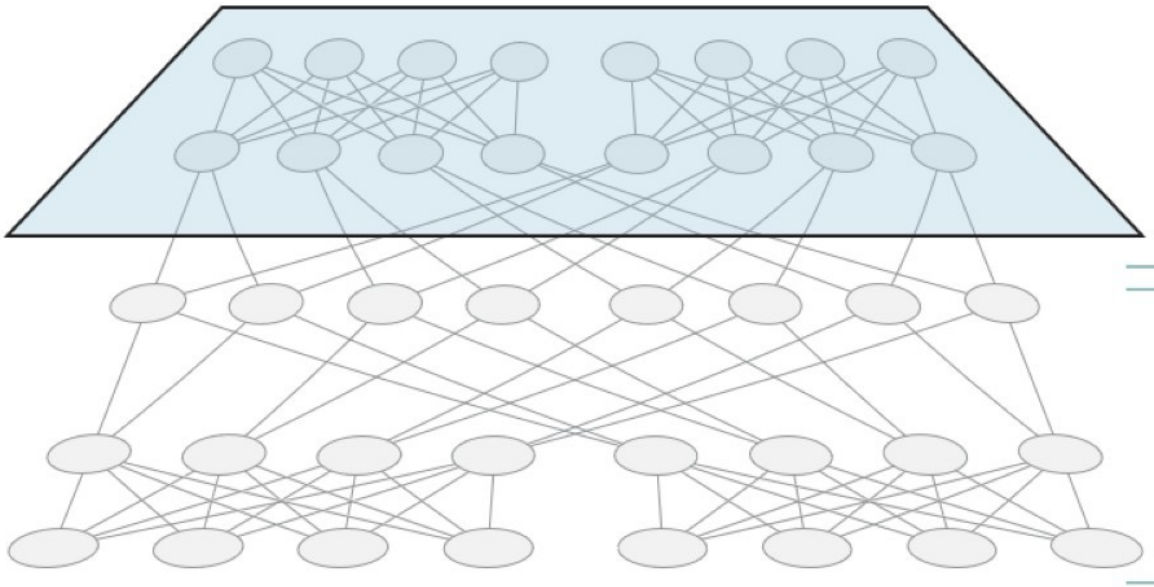


cattle not pets

multiple sources
no strong service ties
no strong ecosystem ties

the magic is in the system
not the devices

OPERATIONAL ABSTRACTION



you need solid operational abstraction

if you're using multivendor here ...

OPERATIONAL ABSTRACTION

some open-source examples

automation abstraction

ANSIBLE

Progress Chef

SALT PROJECT

observability abstraction

ZABBIX

Nagios

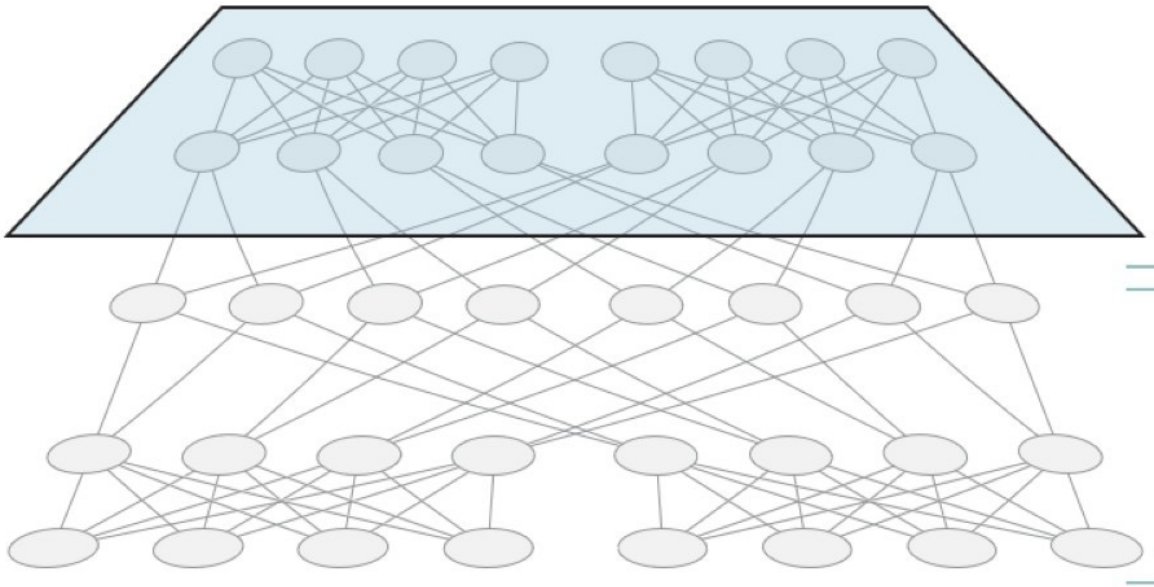
ICINGA

CACTI

PROMETHEUS

OPENCONFIG

OPEN-SOURCE ABSTRACTION



you need strong interoperability here

if you're using multivendor here ...

CONTROL PLANE ABSTRACTION

keystone

complexity

complexity

configuration

versus physical devices

management

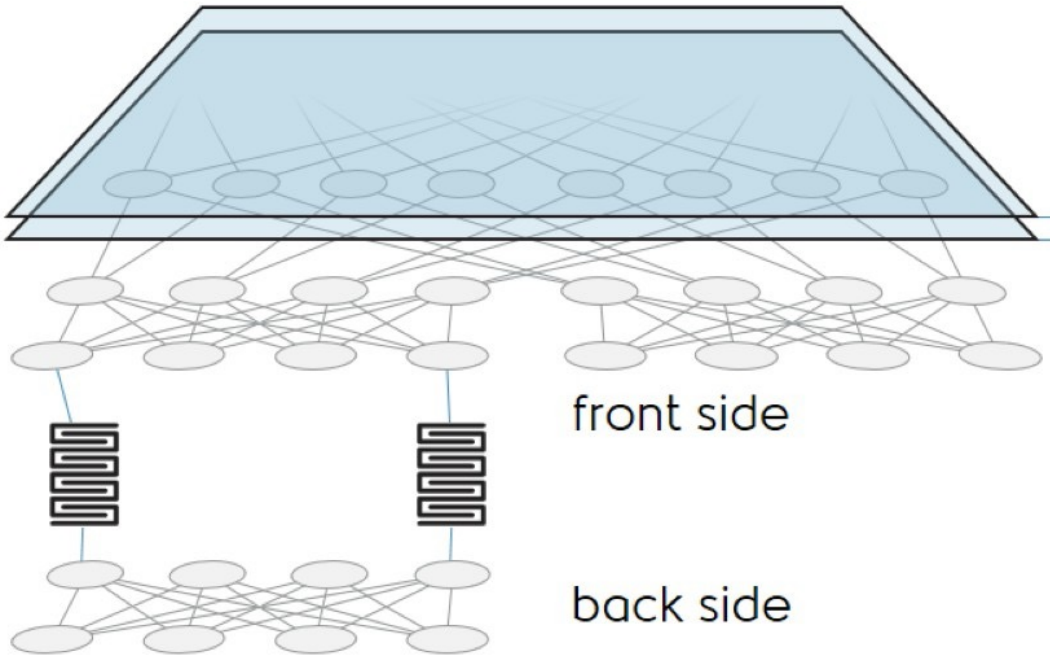
versus physical topology

control plane

versus optimal utilization

versus optimal traffic delivery

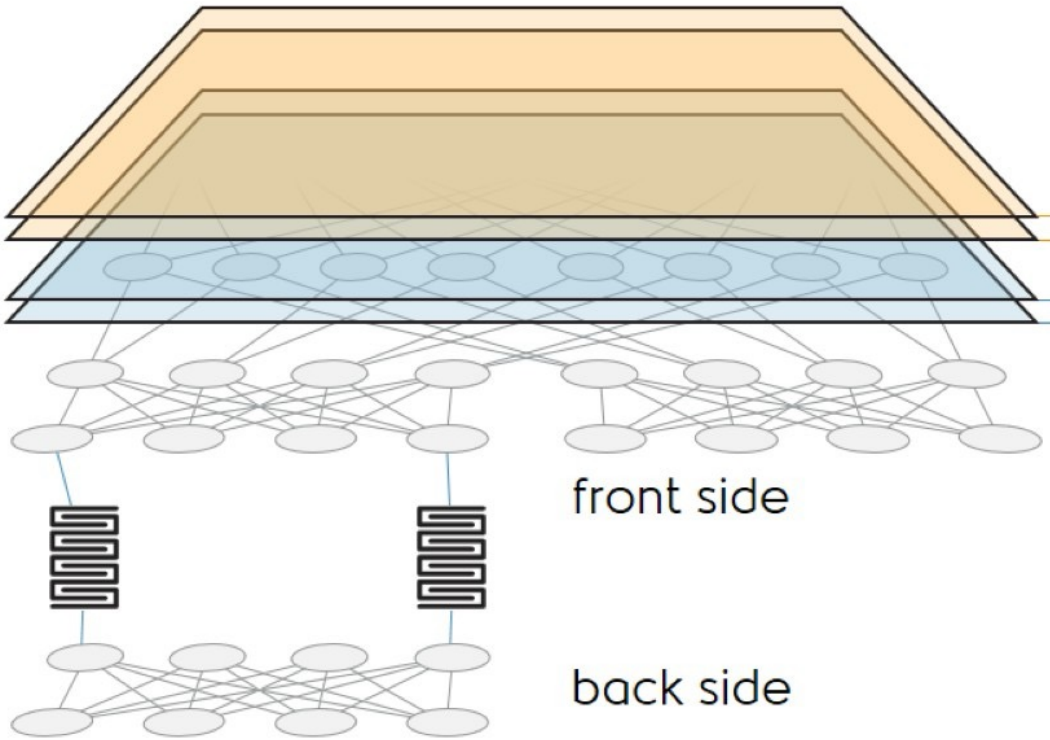
ABSTRACTION



operation resilience

configuration diversity

IDEALIZED ARCHITECTURE



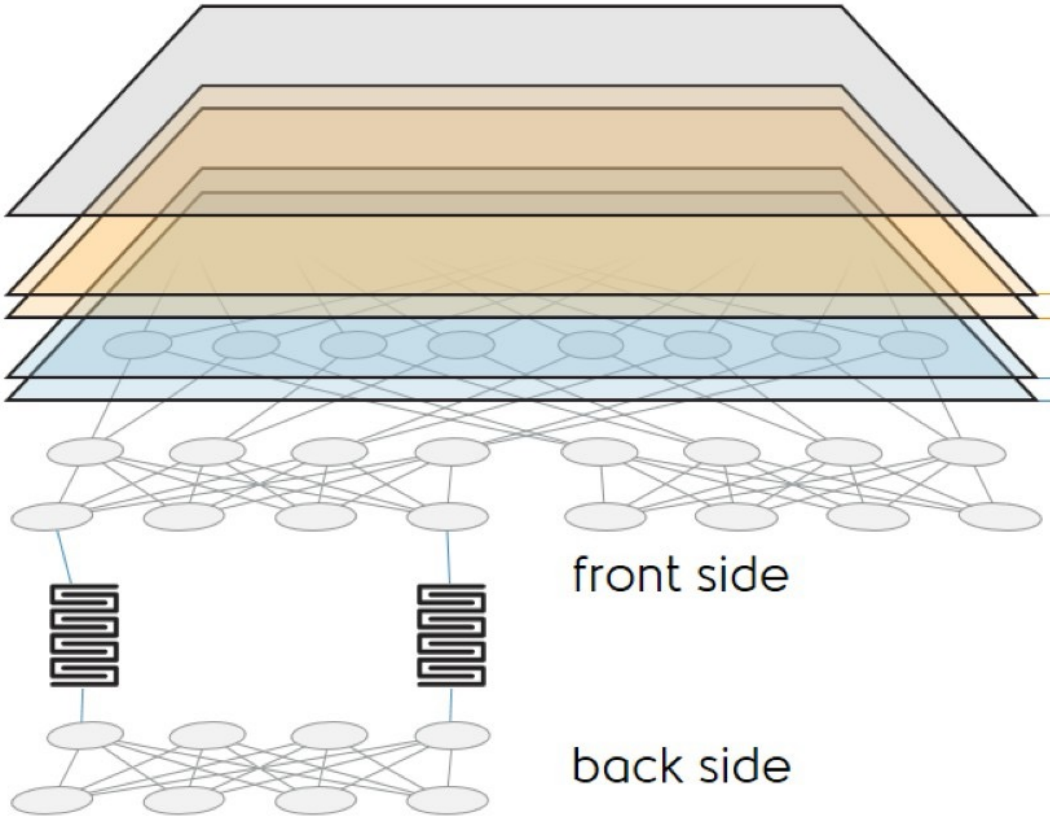
centralized control — policy

distributed control — reachability

operation — resilience

configuration — diversity

IDEALIZED ARCHITECTURE



overlay — intent

centralized control — policy

distributed control — reachability

operation — resilience

configuration — diversity

IDEALIZED ARCHITECTURE

LEAN INTO (EFFICIENT) FLEXIBILITY

we all want “easy to operate”

if you stick with the “easy to operate thing” a vendor creates

- you must rely on that single vendor supply chain

if you want a diversified supply chain

- you must build that “easy to operate thing” yourself

PUTTING IT ALL TOGETHER