Network Digital Twin

Kireeti Kompella
~~Distinguished~~ Chief Engineer
Juniper Networks

Menu (prix fixe, for entire table)

- Definition
- Automation
- Testing
- AI
- Digital Twin
- Conclusion

0. Definition

Three Definitions

A digital twin is a virtual representation of an object or system that spans its lifecycle, is updated from real-time data, and uses simulation, machine learning and reasoning to help decision making -- IBM

https://www.ibm.com/topics/what-is-a-digital-twin

A Digital Twin is a virtual copy of a physical asset that incorporates real-time data captured via sensors. By integrating a multiphysics simulation model, IoT and machine learning, a Digital Twin can enhance the product -- HPE

https://www.hpe.com/psnow/doc/a50004409enw

A **digital twin** is a digital model of an intended or actual real-world physical product, system, or process (a *physical twin*) that serves as the effectively indistinguishable digital counterpart of it for practical purposes, such as simulation, integration, testing, monitoring, and maintenance

https://en.wikipedia.org/wiki/Digital_twin

"… the first practical definition of a digital twin originated from NASA in an attempt to improve the physical-model simulation of spacecraft in 2010 …"

"… called the "digital twin" by John Vickers of NASA …"

-- wikipedia

Yeah, this *is* rocket science!

1. Automation

What are the goals?

How well have we automated?

How can we do better?

Automation: 1. Enhancing Human Capability

Automation: 2. Reducing Human Error

There have been some far-reaching errors in the past year

This is not new

How many unreported incidents occur?

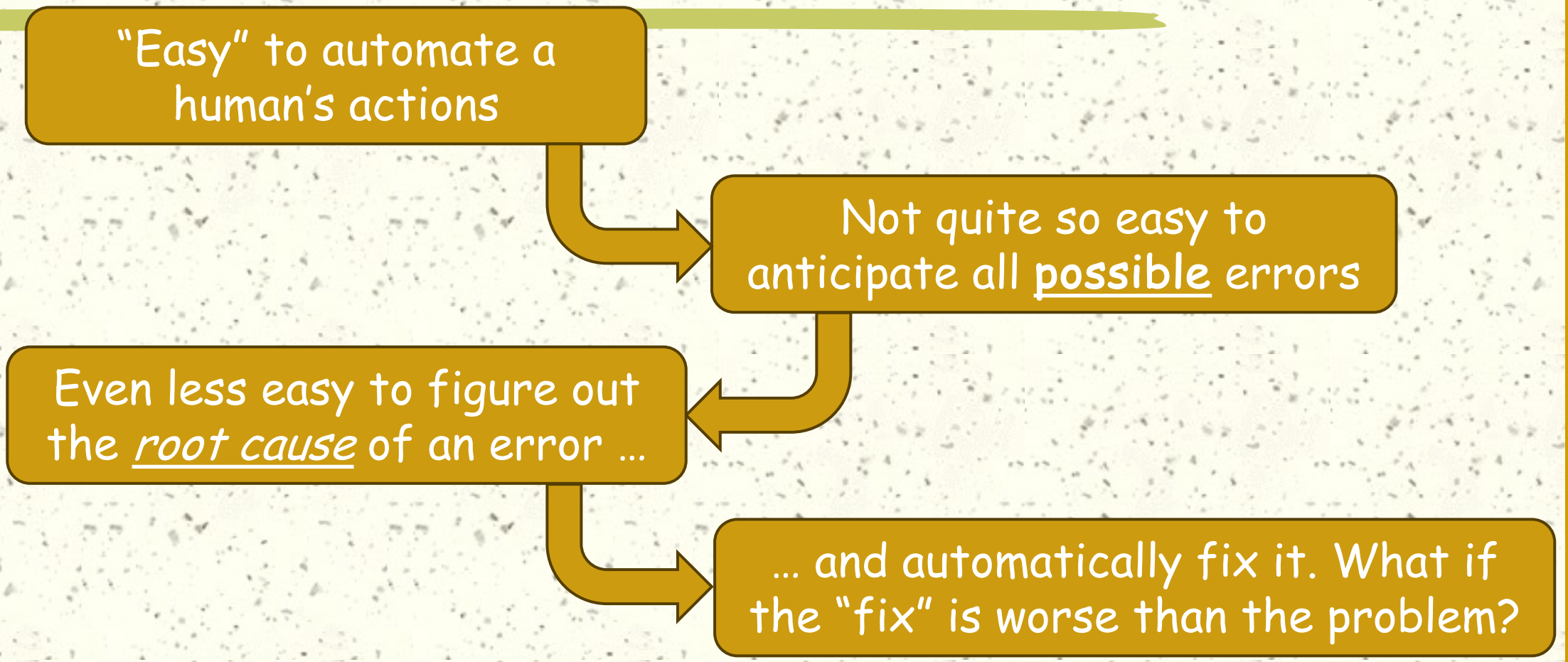
Automation can magnify the damage of an error!

Automation *does* things: faster, more reliably, with fewer errors

...

But can it *find* errors? Can it *fix* things? *Does* it?

3. Is Automation Sufficient?

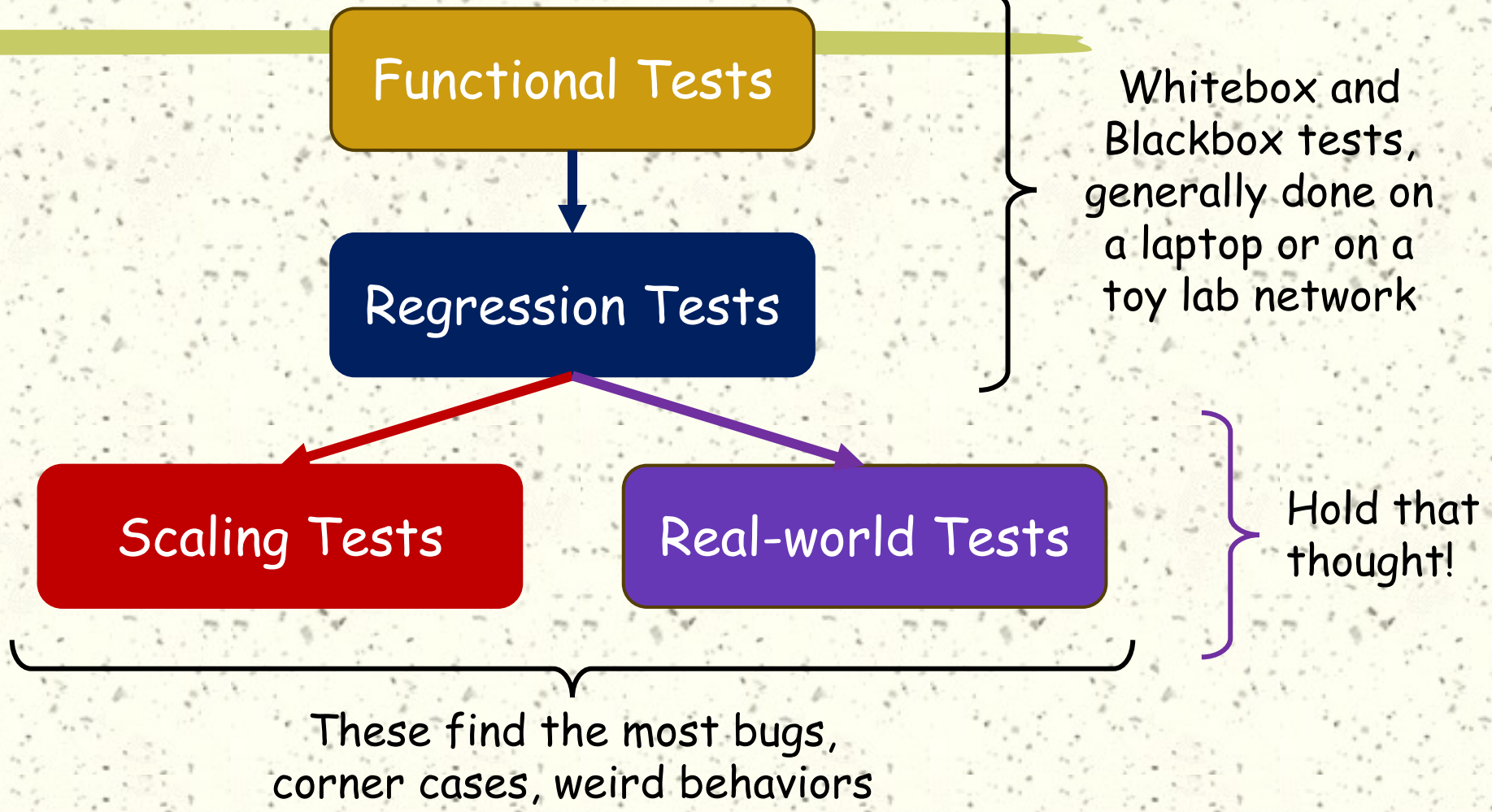


2. Testing

Can't prove automation is correct and error-free …

… but can reduce common causes and thus the probability of error

Types of Tests



3. AI

(yay! our problems are over!)

(Nah! be careful!)

AI (really, ML) Can Help

Find anomalies, unexpected events or human/machine errors

- In configuration
- In telemetry
- In traffic patterns
- …

Find the root cause for the anomaly or error

Suggest (or execute) remedial action

Great!

Oh, Wait! Where's the Data to Train the ML?

Scaling Tests

Real-world Tests

Remember these?

How often does the network break?

How to create controlled situations?

How to recreate a situation for debugging?

4. Digital Twin

Network Digital Twin

NDT allows you to clone your network for …

… recreating a problem, and trying out fixes

… looking at the past, the present, the probable, the possible, the never-to-be

4.1 Network Digital Twin

What Is a Network Digital Twin?

An NDT is a clone of a real network that captures:

- the connectivity (topology)
- the Operating System powering each device
  - the control plane of each device (routing protocols, RIBs, device management, …)
  - the management plane of each device (configuration)
  - APIs to connect to and control each device
- the data plane of each device (ports, interfaces, forwarding, FIBs, ACLs, queues, …)

The nodes and links in the NDT are virtual

- A node in the NDT runs the same OS, control plane and management plane as a "real" node
- A link in the NDT is a virtual link that generally represents a "logical interface"

Traffic in an NDT is artificially created, typically by traffic generators

External routing in an NDT is artificial, typically send by route injectors

Uses of an NDT

Understanding	• A network is a complex system; its behavior is dependent on multiple interacting processes. An NDT can shed light on how real networks behave
Simulation	• There are situations where simulation is the best approach to determining the network reaction, superseding "design", "theory" and "prediction"
Prediction	• ML techniques are being used to predict some events in a network. An NDT can validate such predictions. Can also help fine tune ML algorithms
Education	• An NDT is an excellent vehicle for training NoC personnel on not just the basics of network operation, but on difficult situations which are hard to create in a lab
Experimentation	• One cannot experiment on a live network; nor at scale in a lab. An NDT allows "proper" experimentation, enabling new architectures and new approaches. You can go crazy, knowing you're a reboot away from reset ☺
Data generation	• Training ML for detecting failures would take too long as failures are rare. With an NDT, one can inject known errors, making this more fruitful

Fidelity

A digital twin's utility increases dramatically as its fidelity to real systems increases

Whether one is interested in …

- "Understanding" complex systems
- Simulating possible behaviors
- Prediction
- Education
- Experimentation
- Generating test data for ML

… the results will better reflect actual behavior the more faithful the digital twin is

However, fidelity comes at a cost

One must target fidelity based on desired and potential use cases

4.2 Benefits

Better, More Efficient Research & Development

Data-oriented decision-making	Lower-cost, lower-impact way to experiment	Efficiency improvements
• What if I made *<xyz>* change to my product? • Should I use *approach A* or *approach B?*	• Set up the desired environment • See the results, the trade-offs and the consequences	• Create a controlled environment • Optimize, measure; optimize, measure; optimize, measure; … • … the optimization process itself is more efficient

Troubleshooting and Diagnostics

On a *real* system:

- Debugging a problem can be difficult, intrusive and uncontrolled
- Recreating an issue can create more problems than it solves
- Getting the necessary diagnostics to get to the root cause isn't easy
- Reality gets in the way ☺

On a *digital twin* however:

- You control the environment
- You can recreate a problem; add instrumentation to help debug it
- Once the problem is recreated, you can test alternative solutions
- There are few real-world consequences

Predictive and Proactive Maintenance

If you've designed a system well, there hopefully aren't too many issues …

… which makes it hard to use ML techniques to learn to diagnose and respond to faults

On a digital twin, you can simulate many faults under various conditions and learn

You can also run simulations faster to get behavior trends, thus start predicting

The Environment is under your control in a Digital Twin!

Takeaway:

- More control
- Fewer consequences

4.3 USE CASES

What If You Had A Faithful Replica Of Your Network?

Network Digital Twin

Devices (and software)
Topology (planning)
Data Plane (traffic)
Control Plane (protocols)
Management Plane

"Planner" on steroids

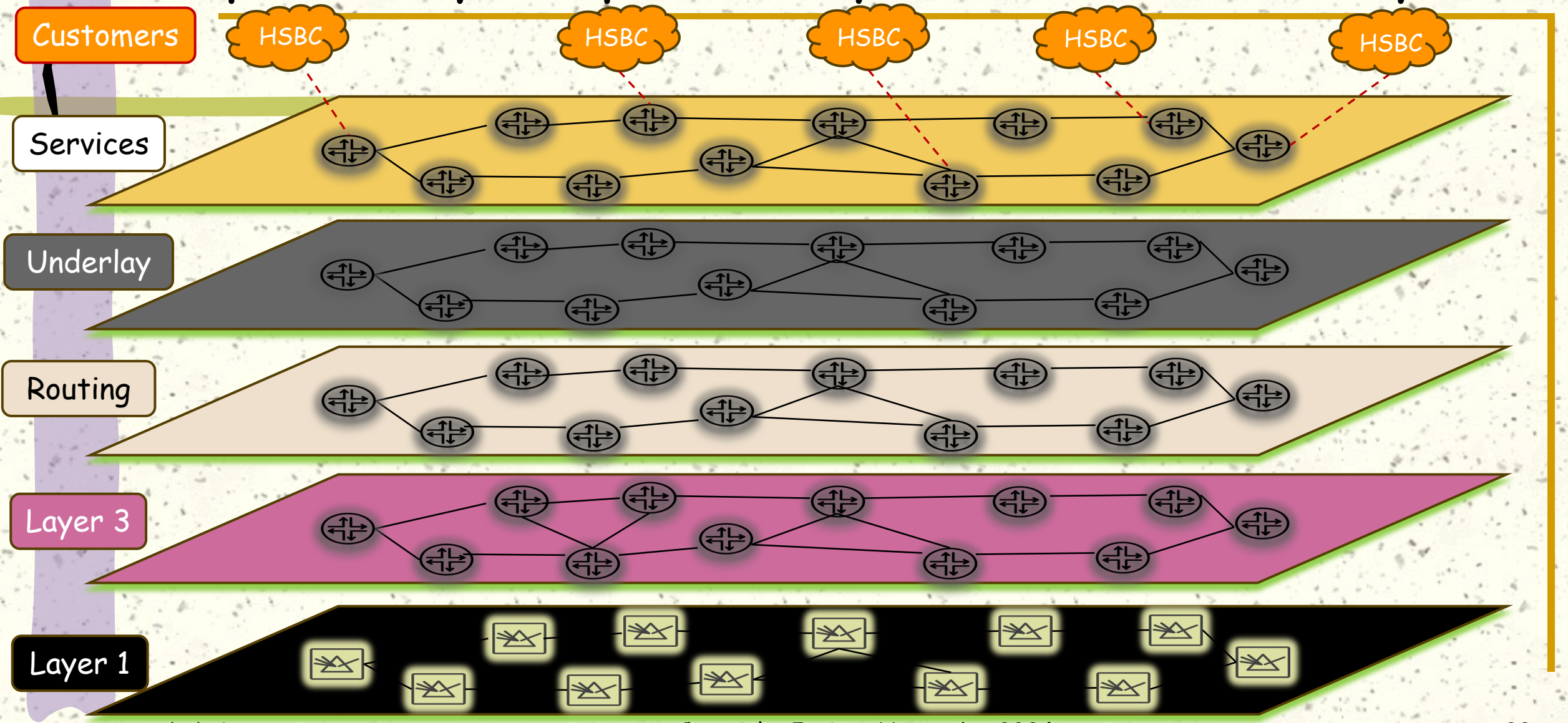
- Platform for training/testing
- Exploring via Mixed Reality?
- "What happened" analysis
- "What if" scenarios

Testing: "CI/CD"-like Pipeline

Software-style discipline for network changes

- Automation as code: snapshot and version
- Commit → run the tests (on the digital twin)
- Success: **deploy** *or* Fail: **roll back**

Dependency Graph: View your Network as Layers



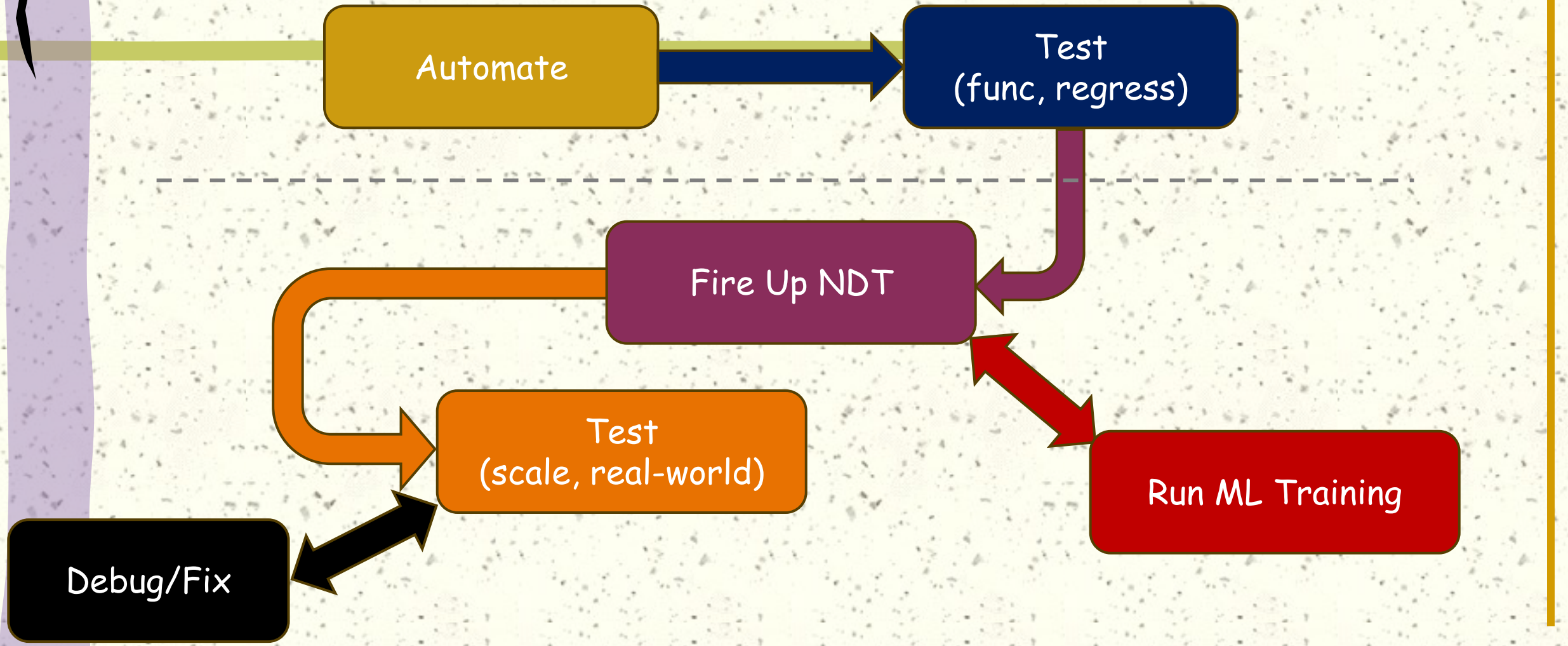
Applications of the Dependency Graph

- Shows you the relationships between devices, routing, underlay, services and customers
- Forms the basis for "Root Cause Analysis" for fast and precise troubleshooting
- Points you to hotspots and potential trouble areas – who will be impacted and how to fix it
- Lets you understand which customers/services will be affected by an outage or maintenance

5. Conclusion

To take Automation to the next phase, we need to invest in Network Digital Twin technology

Network Digital Twins Continue the Story



Network Digital Twin is the Logical Next Step in the Automation Journey (on the path to Self-Driving ☺)

My Biases

- Vendor
- MPLS/Protocols geek (once and forever)
- Automation freak
- Unreasonably idealistic
- Reasonably agnostic

JUNIPER NETWORKS | Driven by Experience™

Thank you!

https://www.juniper.net

kireeti@juniper.net