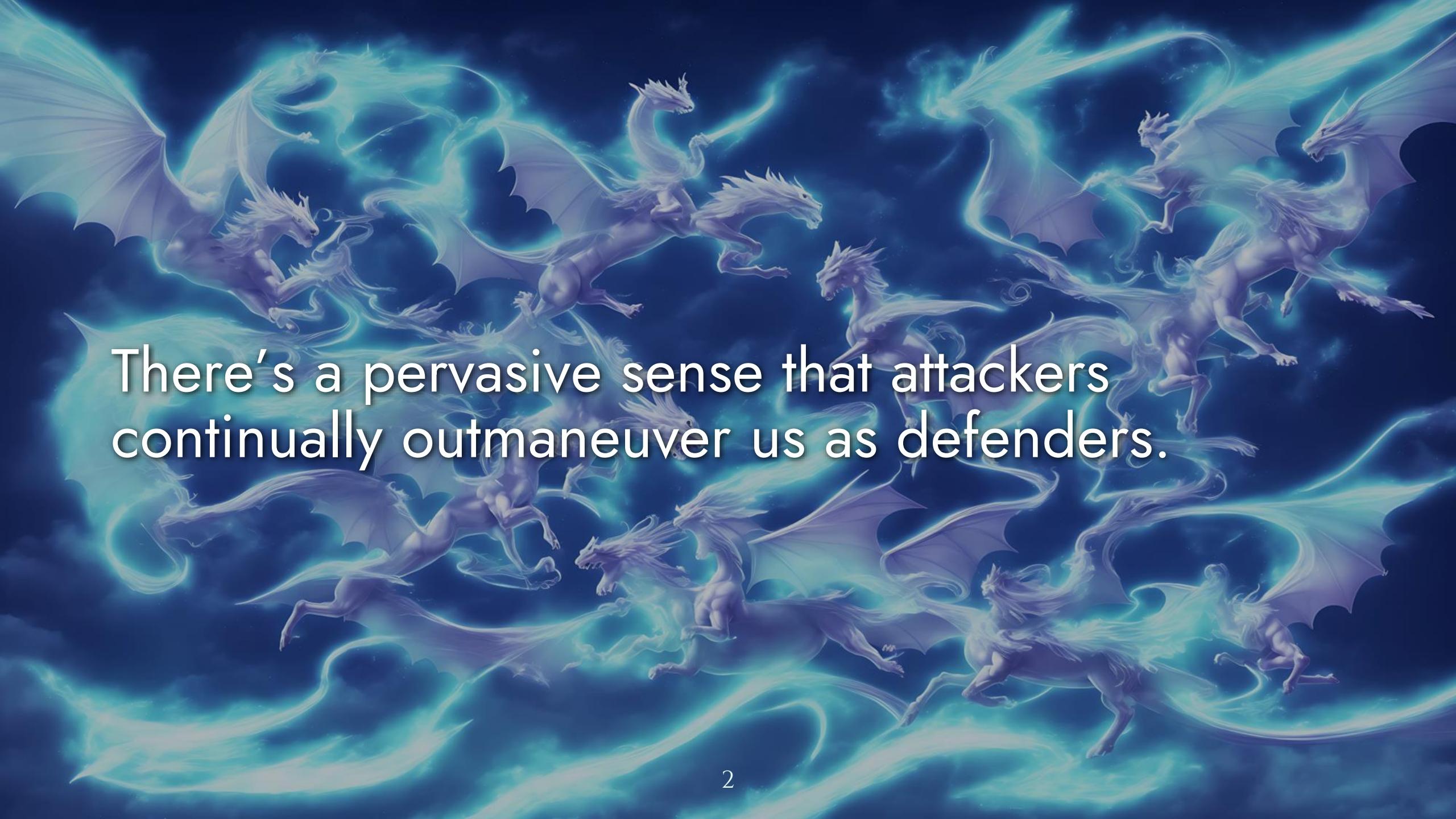




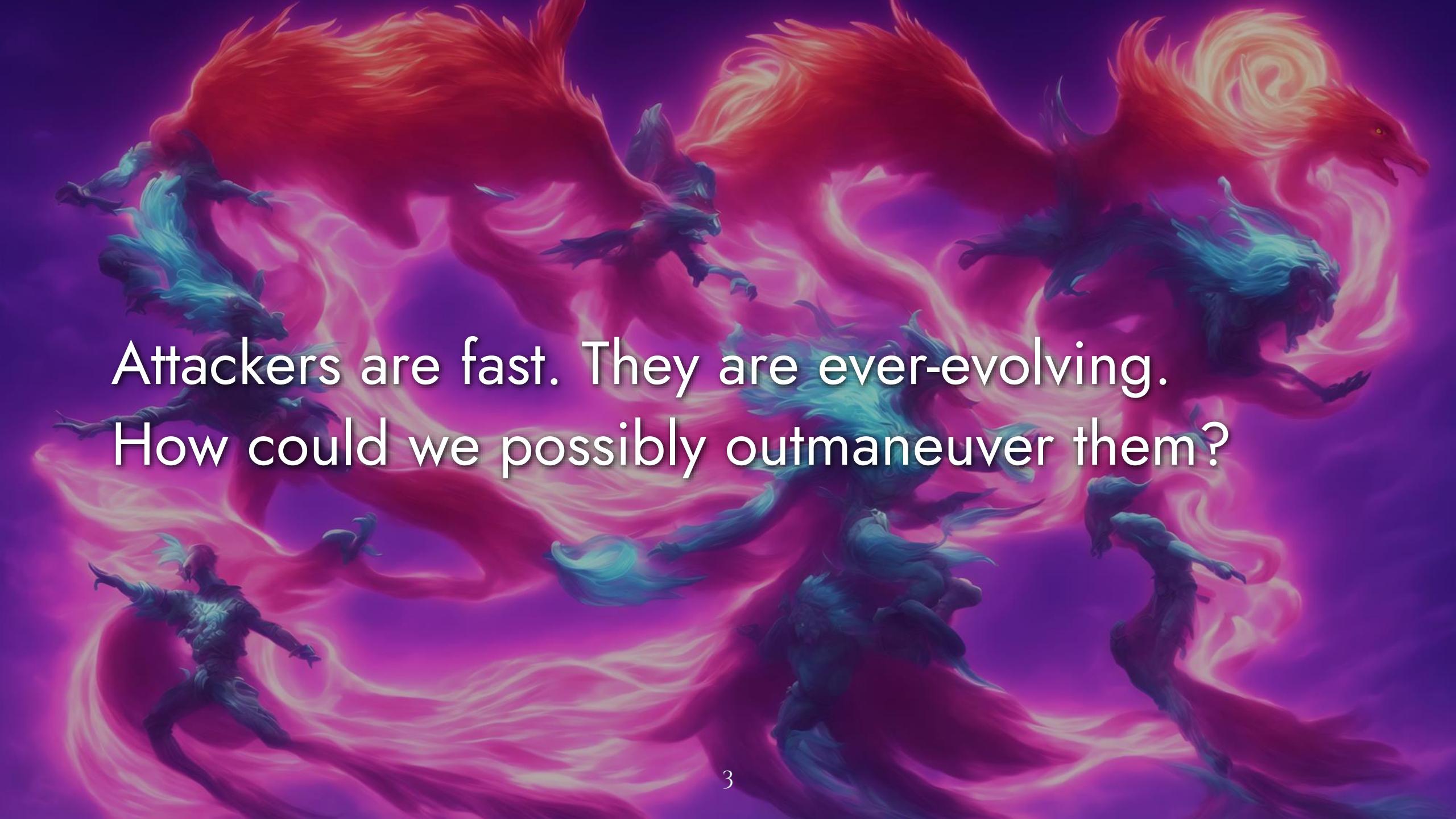
# FAST, EVER-EVOLVING DEFENDERS: THE RESILIENCE REVOLUTION

Kelly Shortridge @swagitda\_ | @shortridge

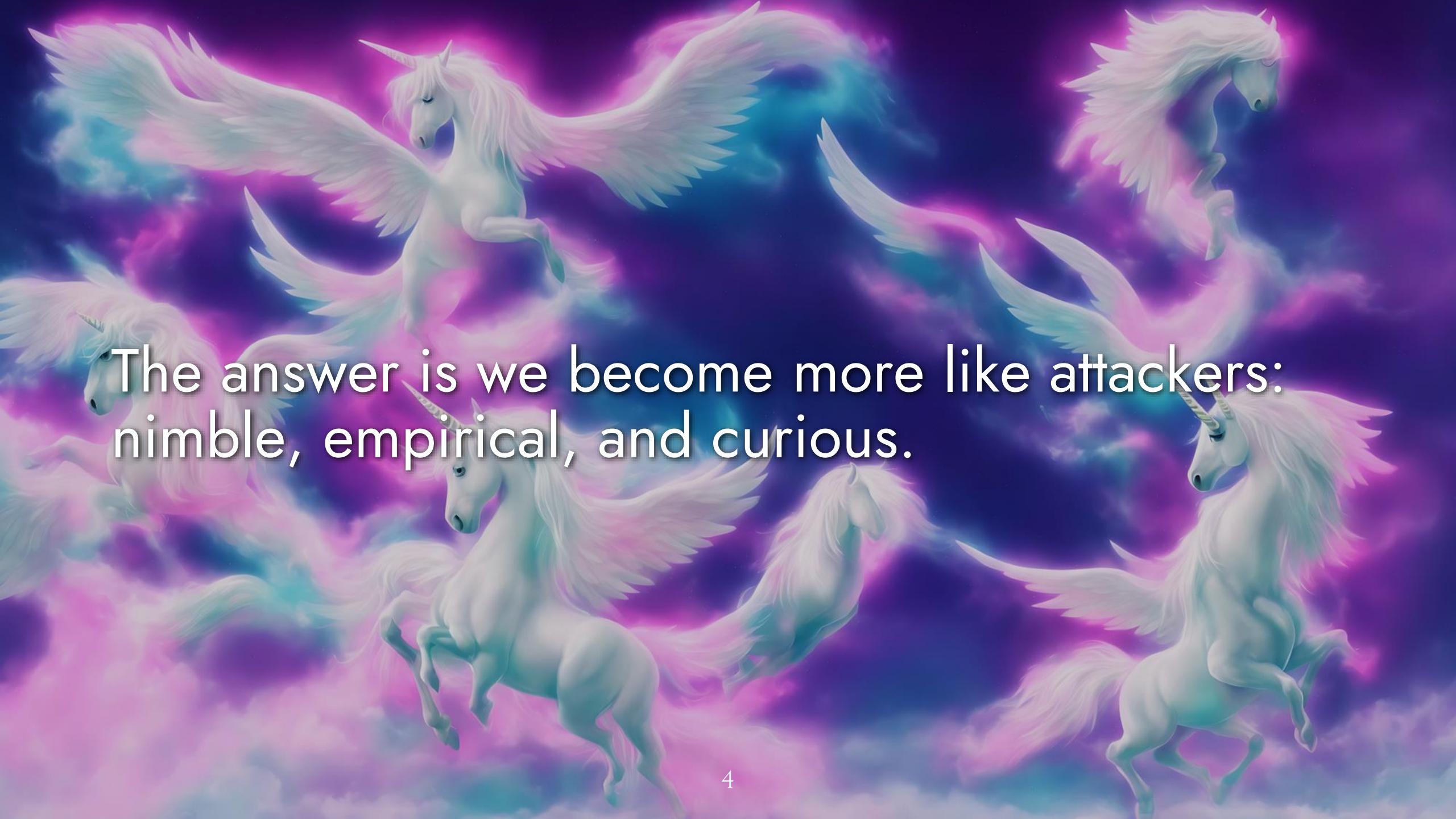
Black Hat USA 2023

A fantastical scene featuring numerous white and light blue winged dragons of various sizes flying through a dark, star-filled sky. The dragons have long, flowing manes and tails, and their wings are translucent with glowing veins. They are scattered across the frame, some flying towards the viewer and others away, creating a sense of dynamic movement and depth.

There's a pervasive sense that attackers  
continually outmaneuver us as defenders.



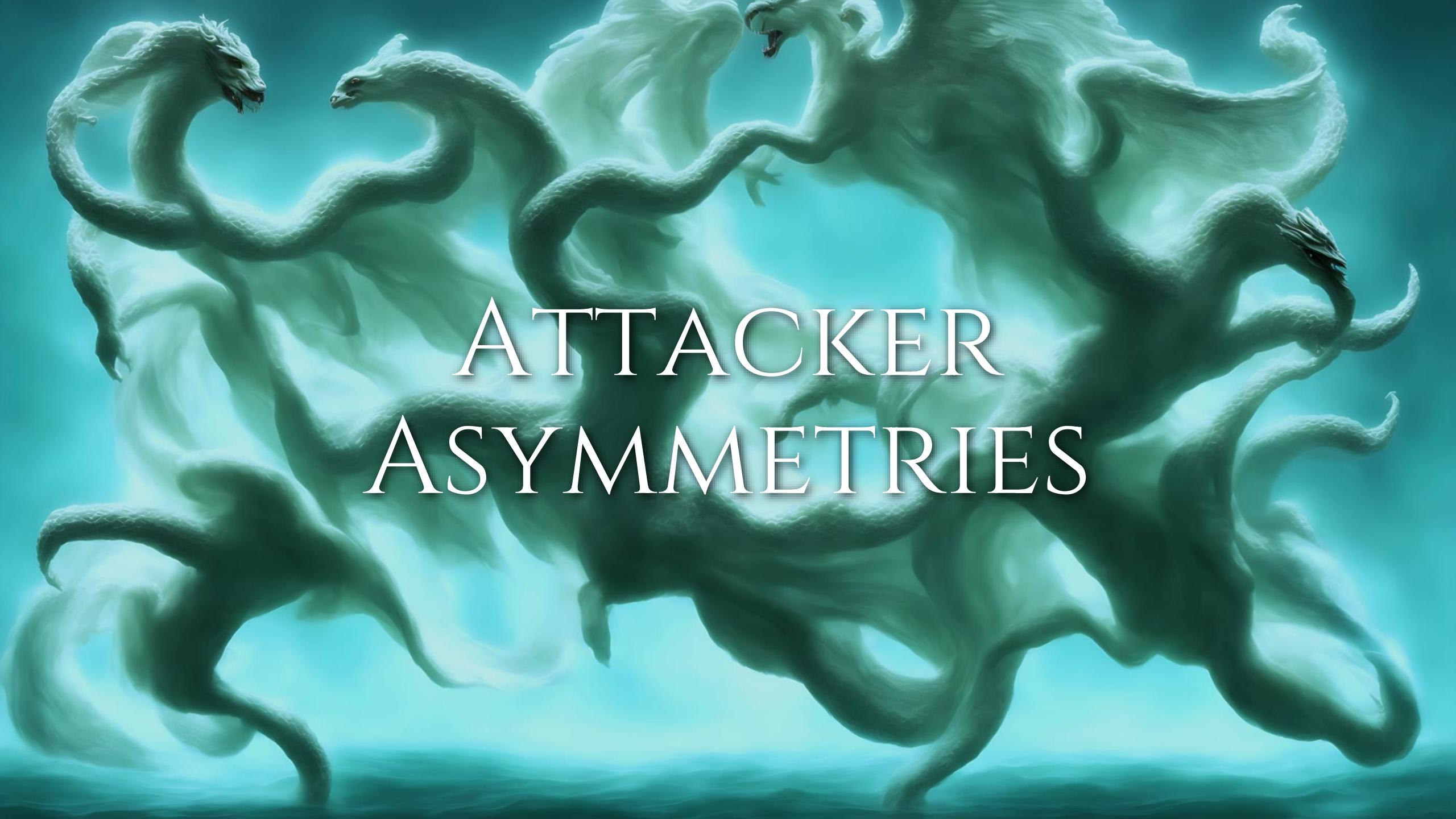
Attackers are fast. They are ever-evolving.  
How could we possibly outmaneuver them?



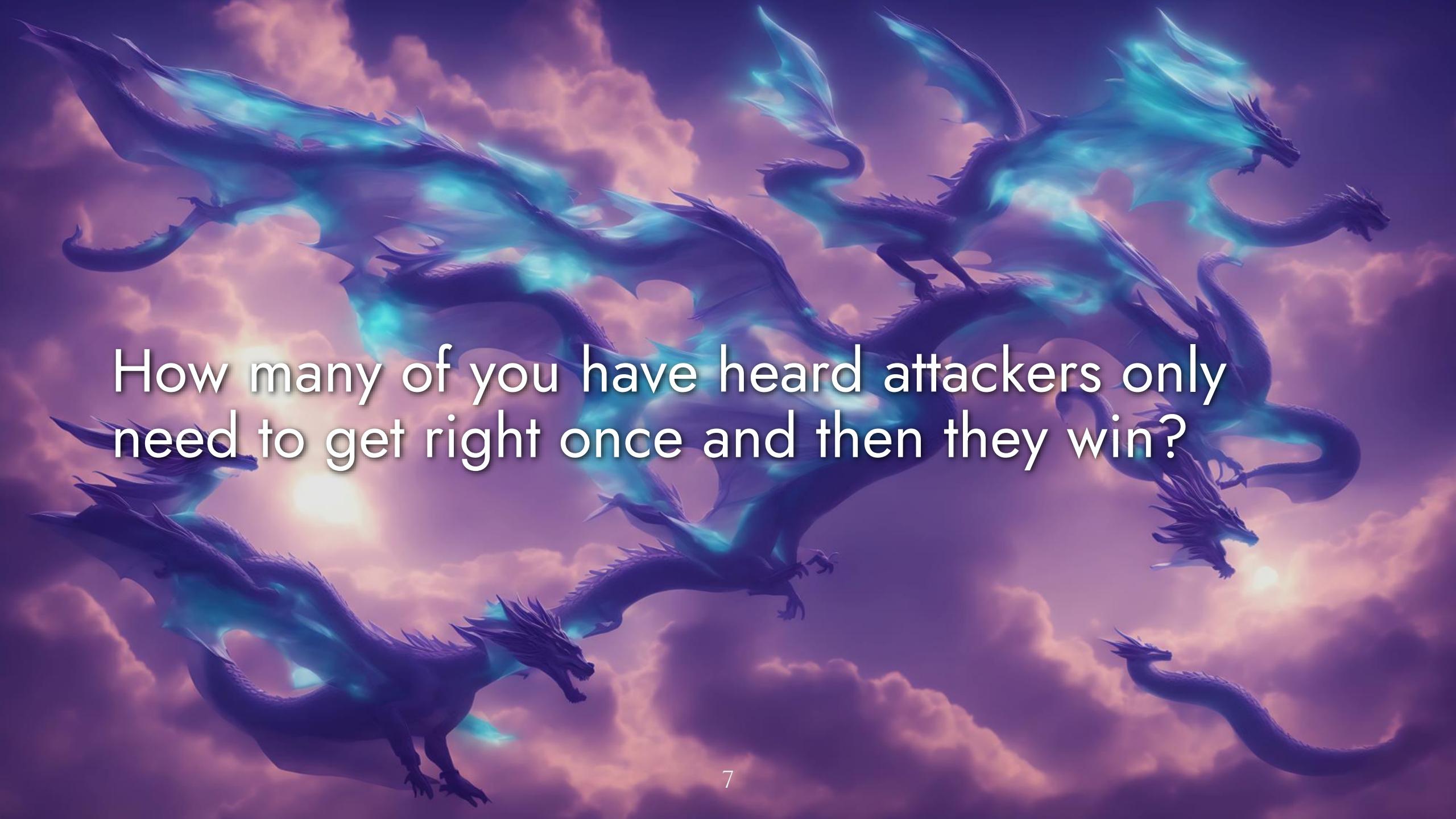
The answer is we become more like attackers:  
nimble, empirical, and curious.



This talk is about revolution – a new paradigm for systems defense, grounded in resilience.

The background of the image is a dark, smoky, and turbulent atmosphere. In the center, several white, multi-headed snakes are depicted in various states of motion. Some snakes have three heads, while others have four or five. They have thick, textured bodies and long, flowing tails. One snake in the upper right is shown with its mouth wide open, revealing fangs. Another in the lower left is coiled tightly. The overall effect is one of chaos, power, and primal energy.

# ATTACKER ASYMMETRIES



How many of you have heard attackers only need to get right once and then they win?

That's a myth. They need to get right once for initial access then get it right *every time* after.

The background of the slide features a fantastical scene with four large, blue, winged dragons. They are depicted in various poses of flight against a backdrop of dark, swirling clouds. The dragons have long, flowing manes and tails, and their wings are spread wide, catching the light. The overall atmosphere is dynamic and mythical.

So, what are attackers' *real* advantages?

1) Attackers have a faster operational tempo

2) Attackers design, develop, and operate mechanisms to outmaneuver us

3) Attackers research interconnections and interactions in systems

4) Attackers have more tangible and actionable success metrics



There is no reason why we can't steal these advantages for ourselves as defenders.

All of these reflect a foundation of resilience:  
the ability to prepare for, recover from, and  
adapt to adverse events.

The background features a vibrant, multi-colored nebula in shades of purple, blue, and pink. Four winged figures are depicted in flight against this celestial backdrop. Three of the figures are white with large, translucent wings, while one figure on the right is dark with large, solid wings. The figures appear to be in various poses of flight or transformation.

We can seize opportunities that grant us these same advantages via the resilience revolution.



# I. FASTER TEMPO

Attackers pivot quickly in the face of adversity.



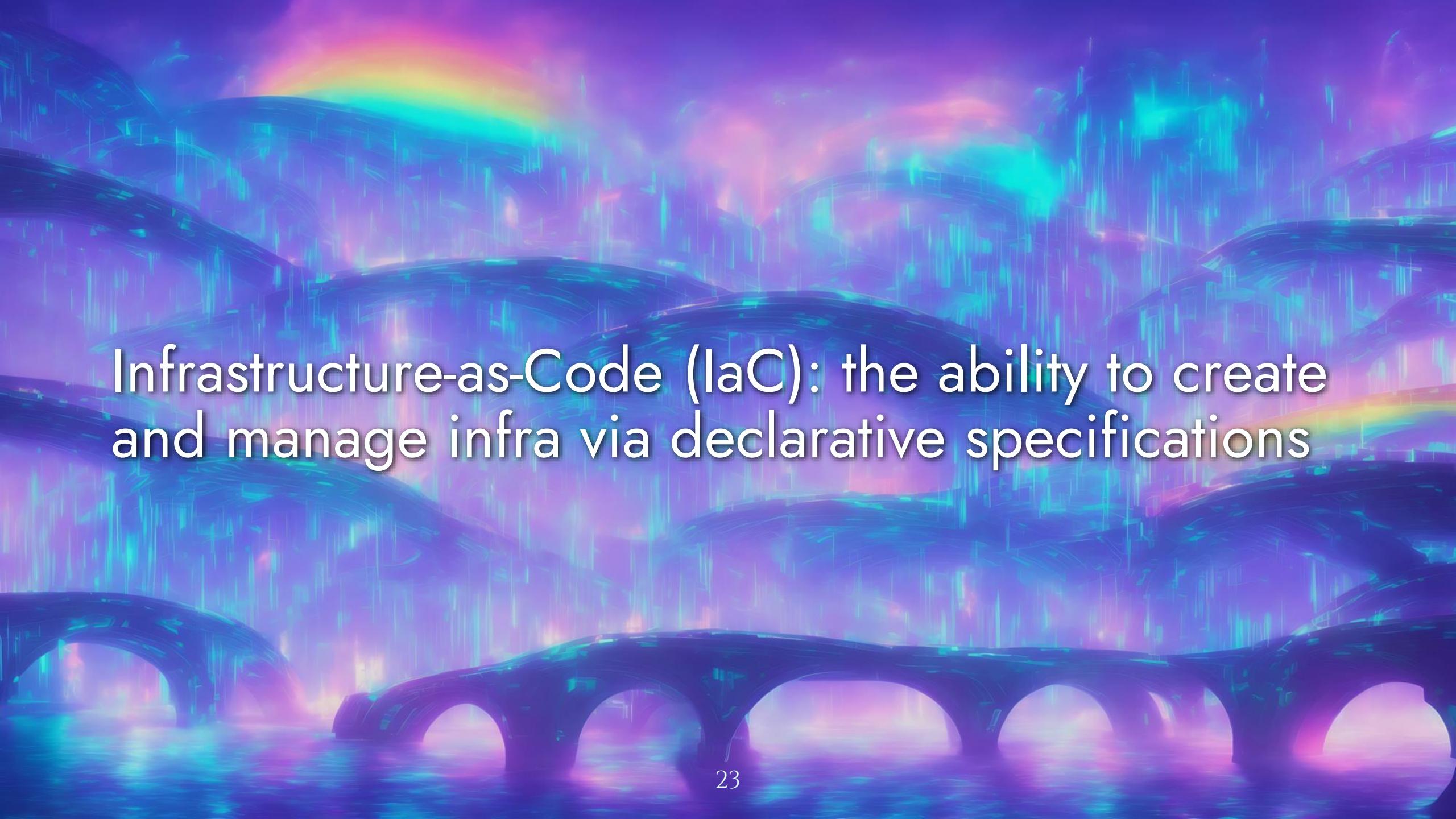
Attackers also rapidly evolve their methods.

We can achieve a faster tempo by adopting approaches from modern software engineering.

The background is a vibrant, futuristic cityscape at night. The scene is filled with tall, translucent buildings that glow with a mix of blue, green, and pink neon lights. The sky above is a deep purple, and the overall atmosphere is one of a high-tech, advanced civilization.

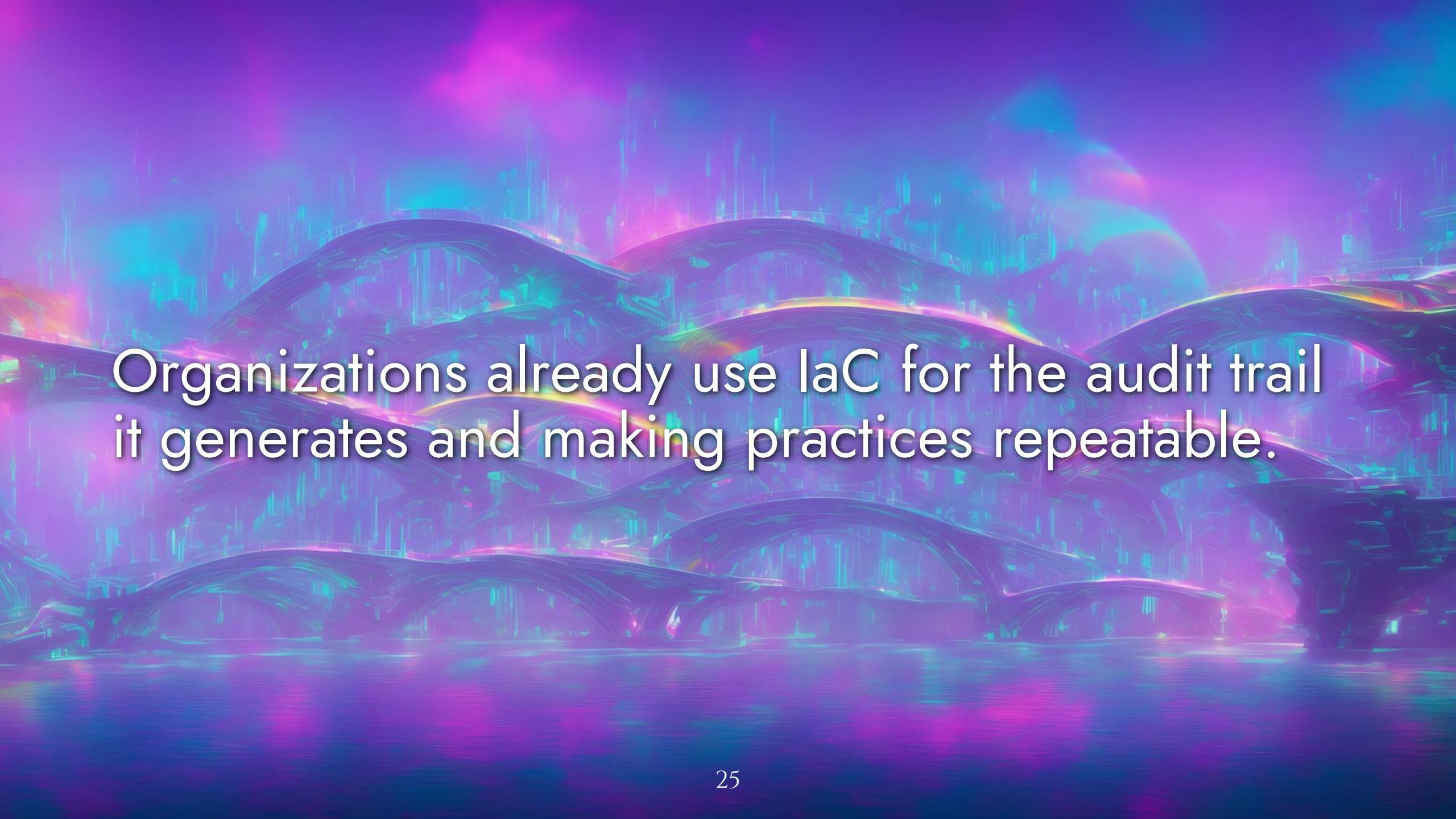
CONFIGURATION AS CODE

CaC: the practice of declaring configurations through markup rather than manual processes



Infrastructure-as-Code (IaC): the ability to create and manage infra via declarative specifications

We generate the same environment every time,  
creating more reliable and predictable services.

The background of the slide features a vibrant, abstract digital artwork. It consists of numerous thin, glowing lines of various colors—predominantly shades of blue, green, and pink—that curve and overlap to create a sense of depth and motion. The overall effect is reminiscent of a futuristic cityscape or a complex neural network visualization.

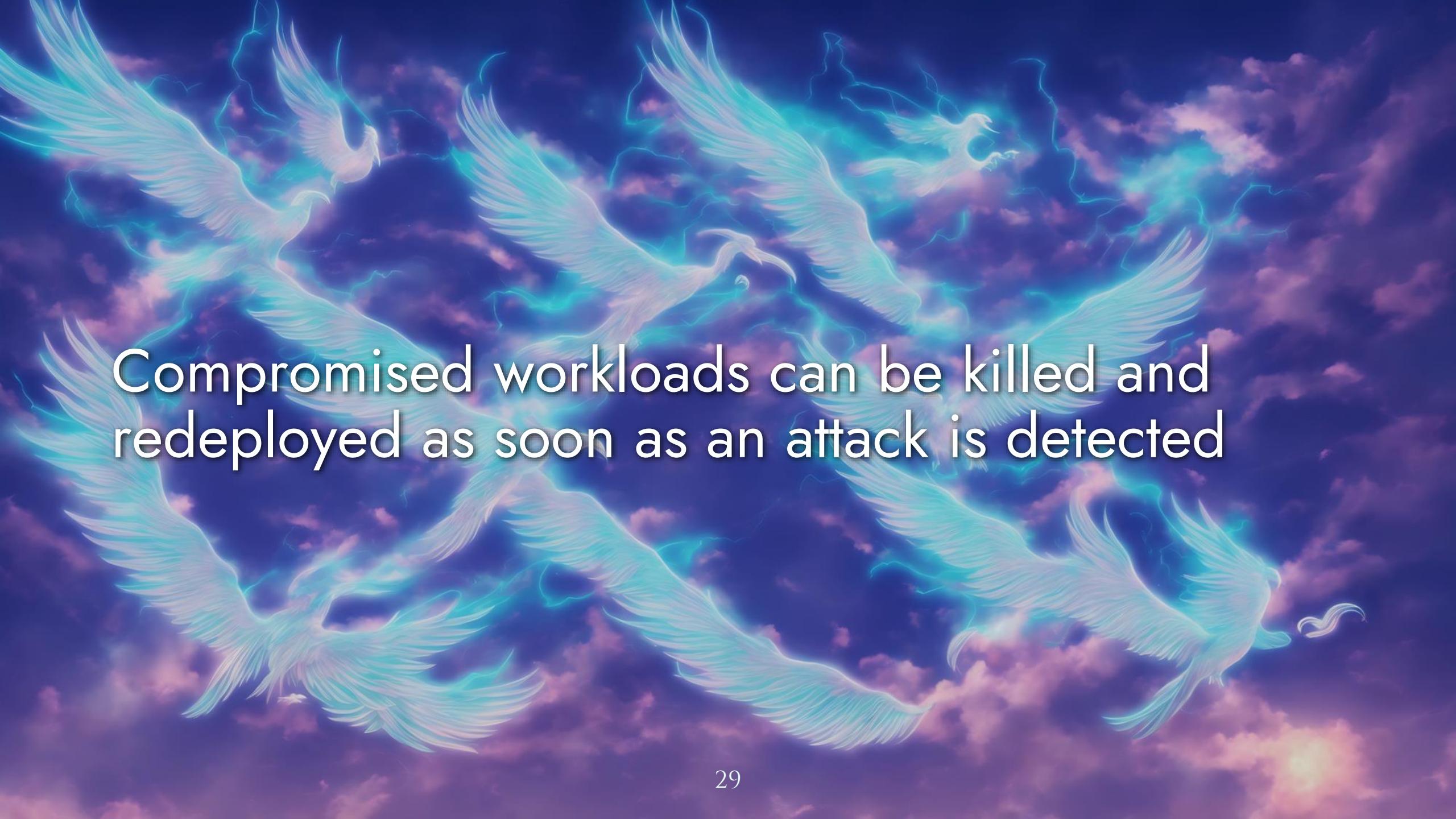
Organizations already use IaC for the audit trail it generates and making practices repeatable.

Let's take a whirlwind tour of laC's bountiful benefits for security programs:

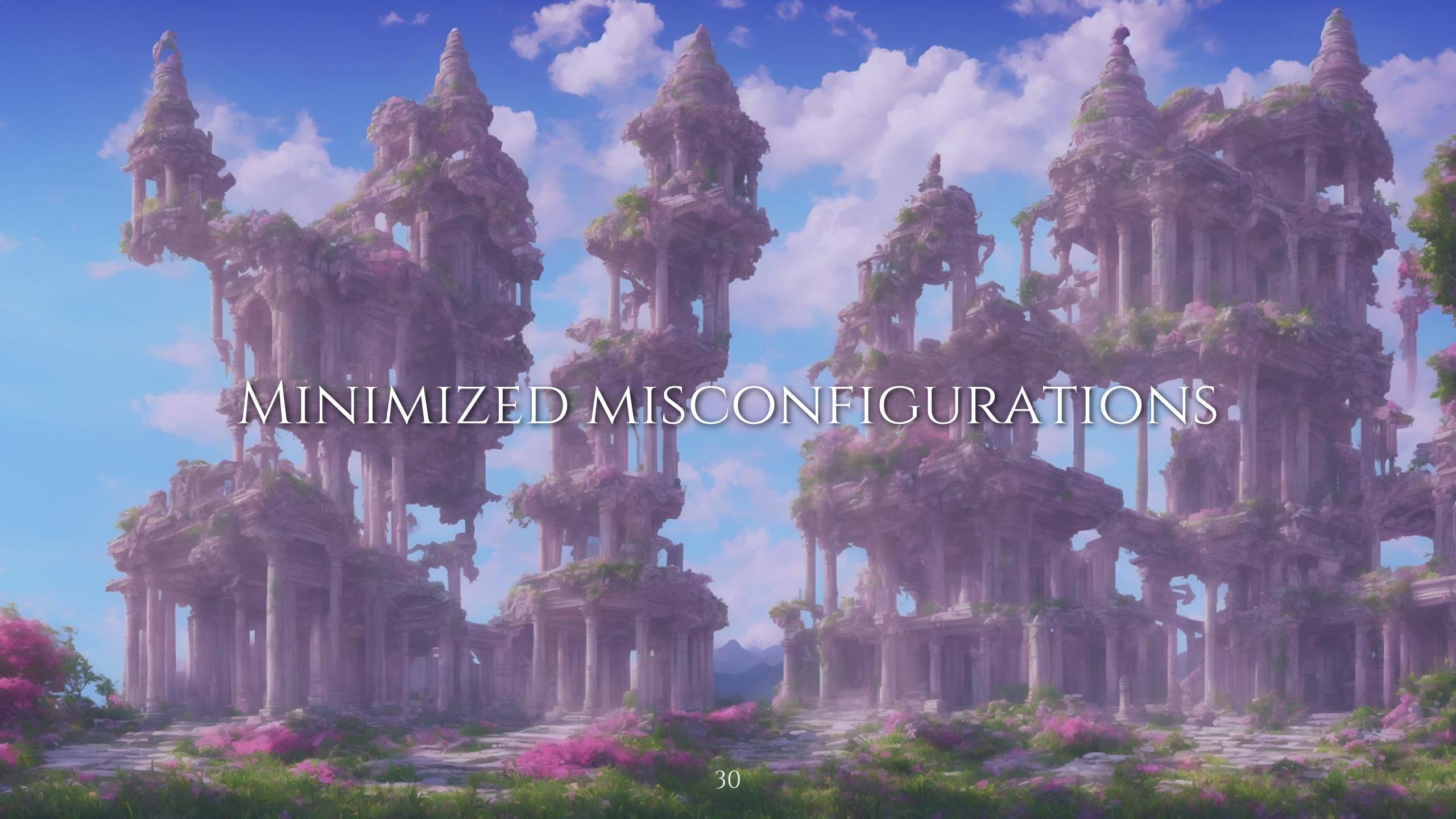


# FASTER INCIDENT RESPONSE

Automatically redeploy infrastructure when  
incidents happen... or even leading indicators



Compromised workloads can be killed and redeployed as soon as an attack is detected



# MINIMIZED MISCONFIGURATIONS

NSA: misconfigurations are the most common  
cloud vuln; easy to exploit + highly prevalent



IaC helps correct misconfigurations by users  
and automated systems (machines) alike



# FASTER PATCHING AND SECURITY CHANGES

The *real* lesson of Equifax: patching processes must be usable, else procrastination is rational

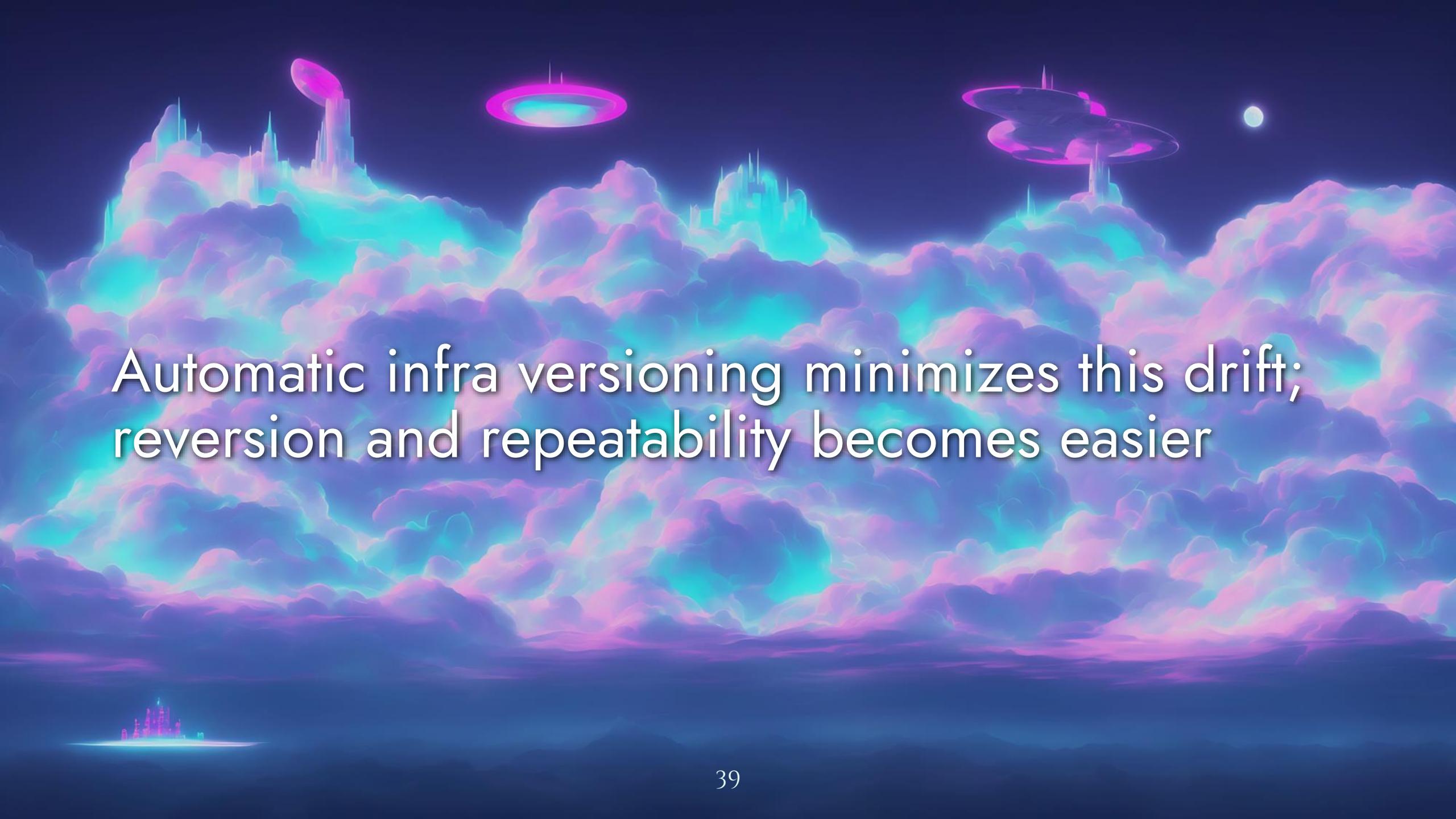
IaC reduces friction for releasing patches,  
updates, or fixes & decentralizes the process

Protip: if an organizational process is unusable or cumbersome, it will be circumvented.



MINIMIZED  
ENVIRONMENTAL  
DRIFT

Environmental drift: configs or other attributes  
“drifting” into an inconsistent state

The background features a fantastical, colorful landscape under a dark sky. In the foreground, there's a small glowing city at the water's edge. The middle ground is filled with large, billowing clouds colored in shades of pink, purple, and blue. Three glowing, translucent rings of light (resembling flying saucers or energy门) are suspended in the air above the clouds. A small, pale moon is visible in the upper right corner.

Automatic infra versioning minimizes this drift;  
reversion and repeatability becomes easier



# CATCHING VULNERABLE CONFIGURATIONS

Status quo is authenticated scanning in production, which introduces new attack paths



IaC removes that hazard, instead scanning the code files to find vulnerable assets or configs



# STRONGER CHANGE CONTROL

IaC introduces change control via SCM,  
enabling peer reviews on configs + changelog

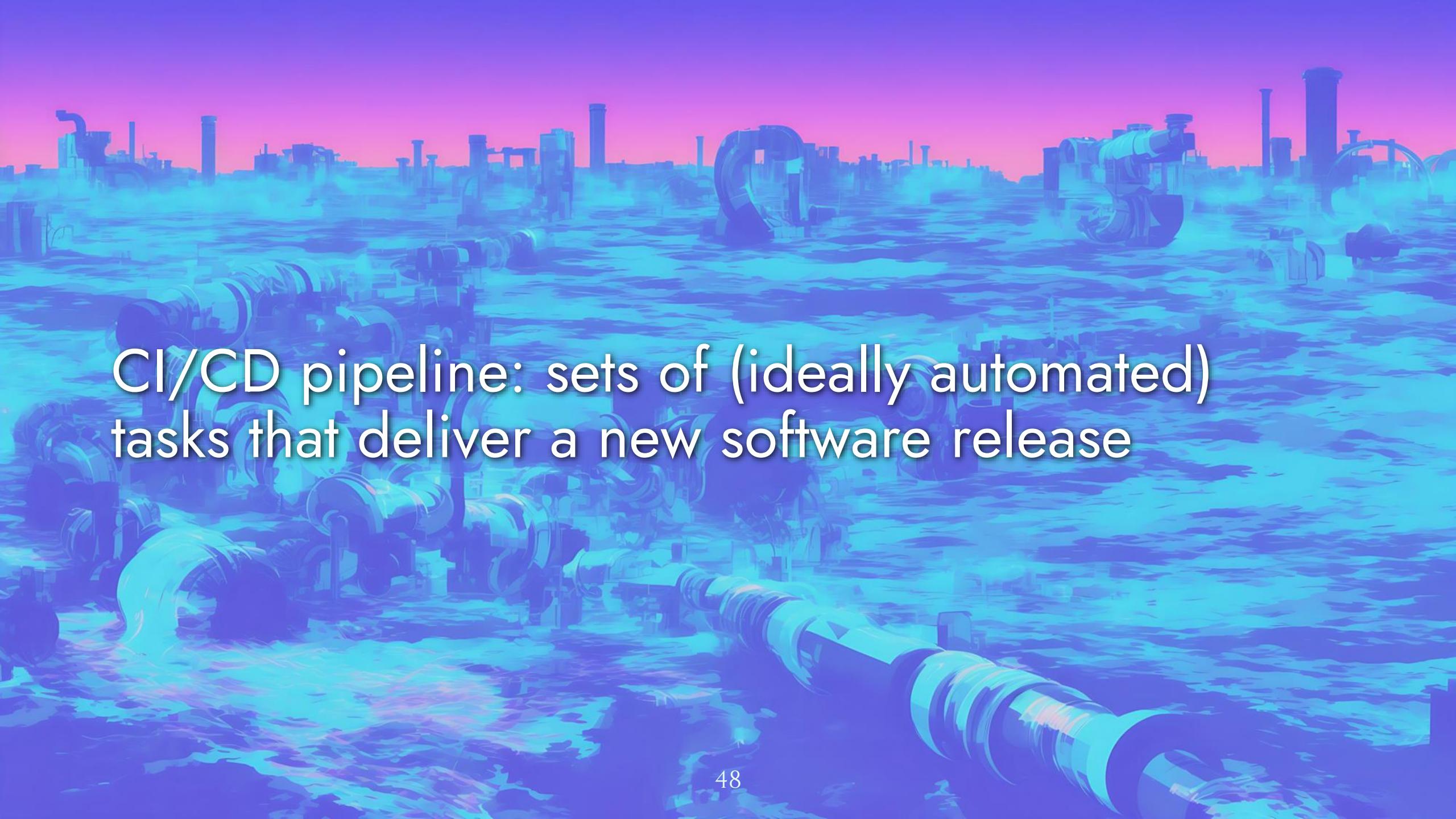


tl;dr laC grants us a faster operational tempo in  
a variety of dimensions



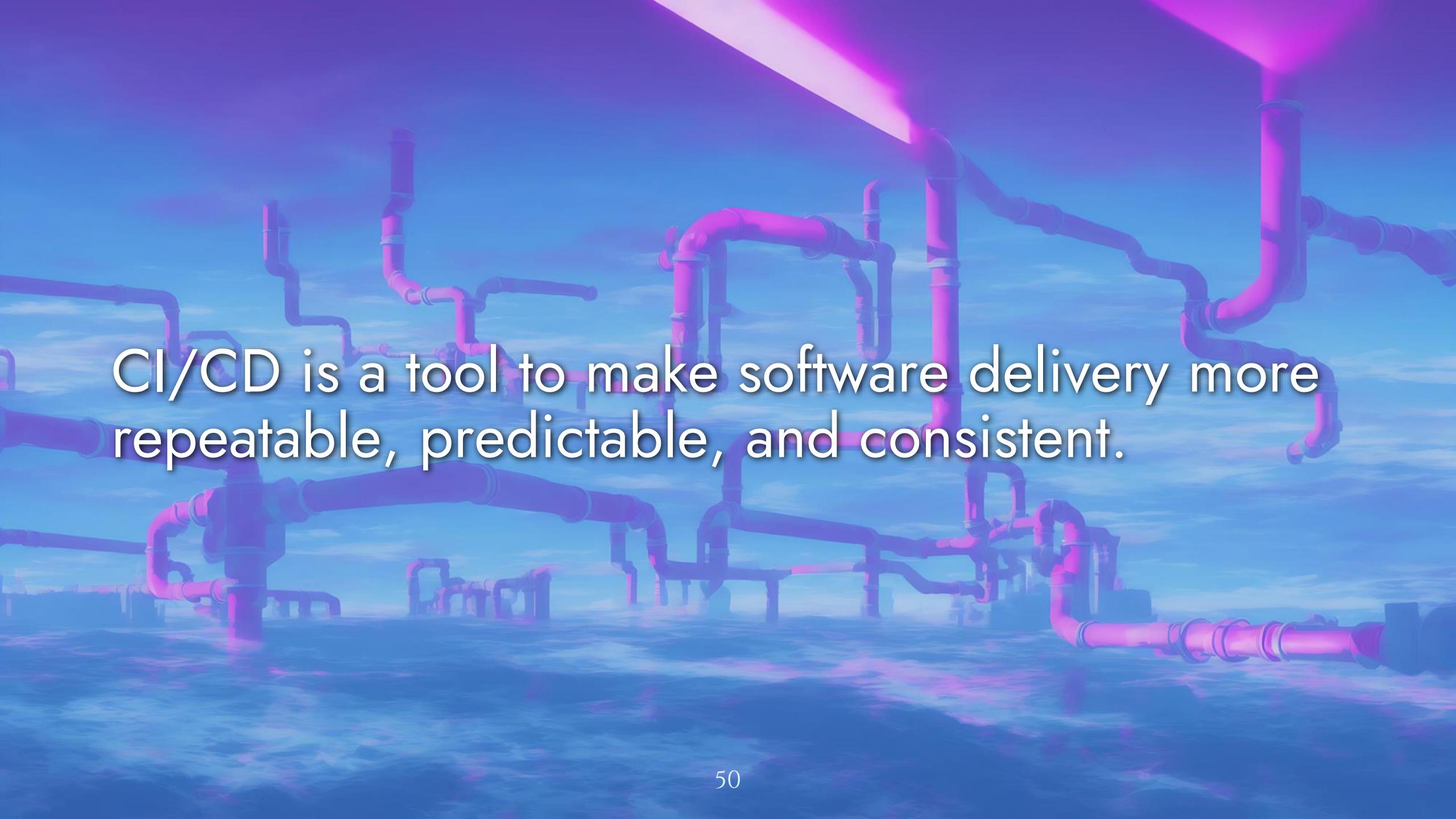
# AUTOMATING SECURITY CHECKS

CI/CD accelerates dev and delivery of software features without hurting reliability or quality

The background features a stylized, low-poly illustration of an industrial landscape. It includes various pipes, valves, and cylindrical structures, all rendered in dark blue and black against a vibrant sunset or sunrise sky with shades of pink, orange, and yellow. The perspective is from a low angle, looking across a field of pipes towards a distant city skyline.

CI/CD pipeline: sets of (ideally automated) tasks that deliver a new software release

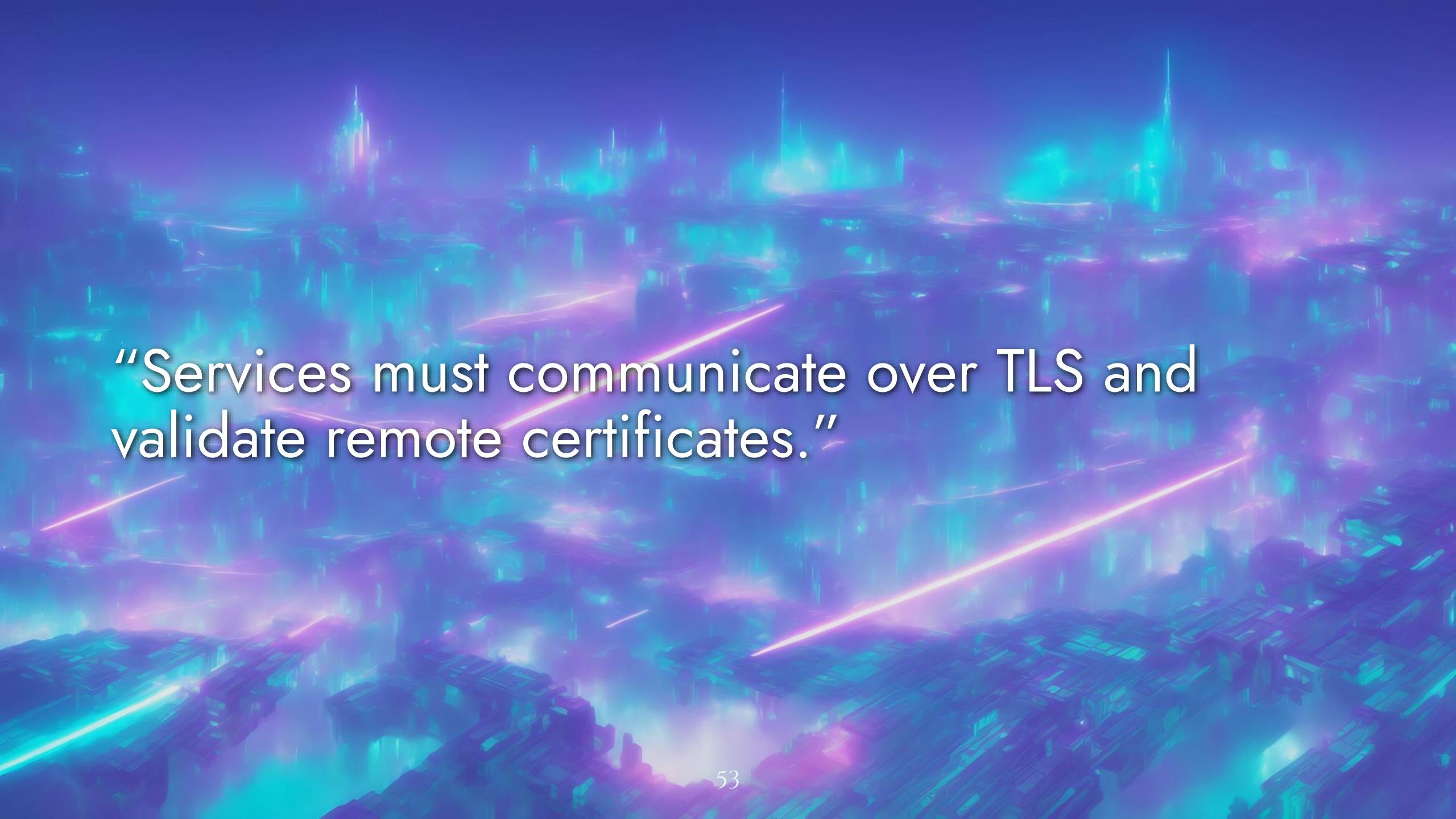
Compiling the app (building) + testing code +  
deploying to test/staging + delivering to prod

The background of the slide features a dense network of pink and purple industrial pipes against a blue sky with white clouds. The pipes are interconnected in a complex web, symbolizing the flow and delivery of software. They are set against a bright, slightly overexposed sky.

CI/CD is a tool to make software delivery more repeatable, predictable, and consistent.

We can enforce invariants: achieve properties we want every time we build + deploy + deliver

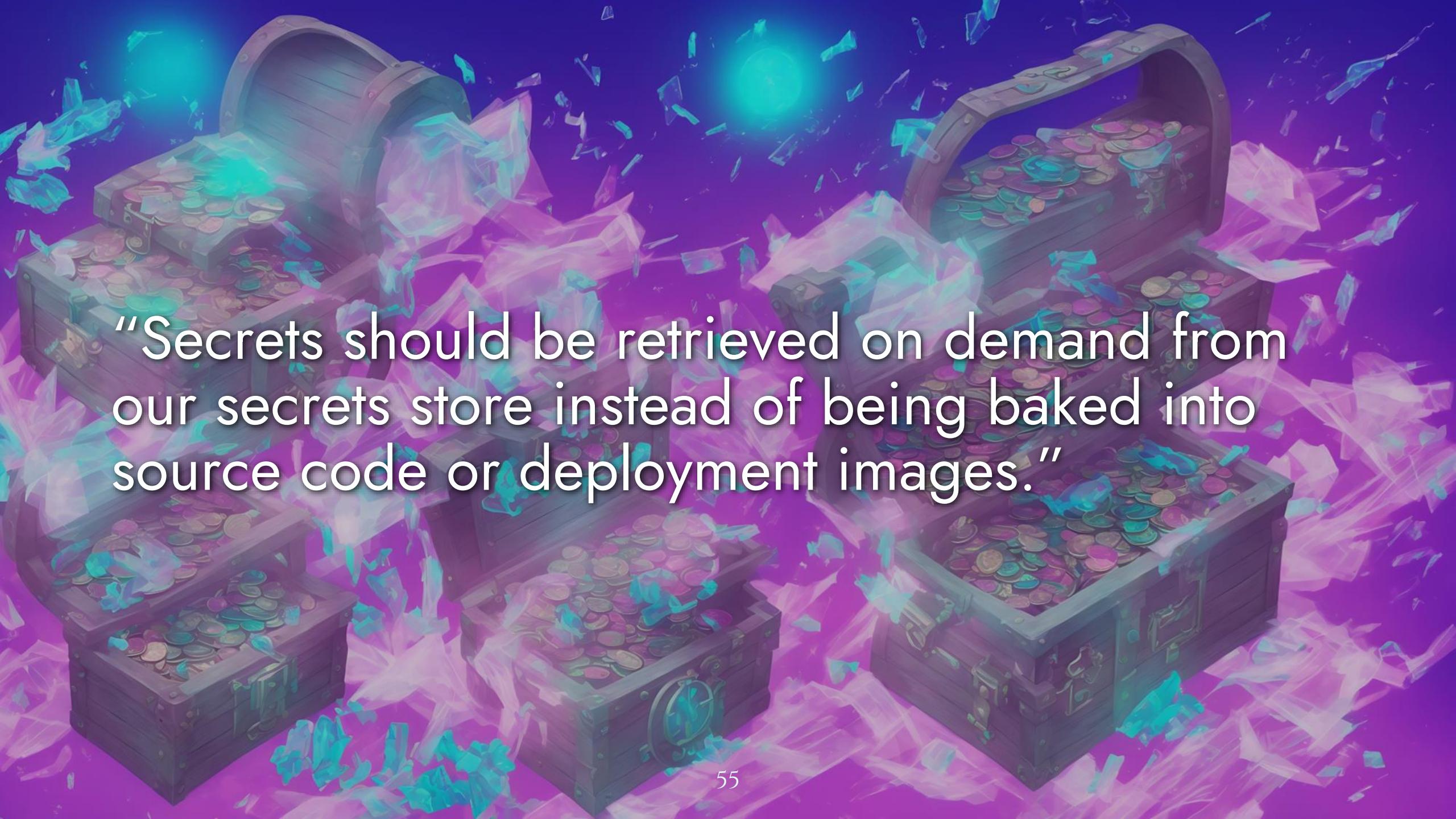
“Database servers should only make outgoing network connections to their replication peers and a short list of core services.”

A futuristic cityscape at night, viewed from an elevated angle. The city is densely packed with buildings that glow with a deep blue and purple light. Streaks of light, resembling laser beams or high-speed traffic trails, cut across the scene, creating a sense of motion and digital connectivity. The overall atmosphere is dark and mysterious, with the city's lights being the primary source of illumination.

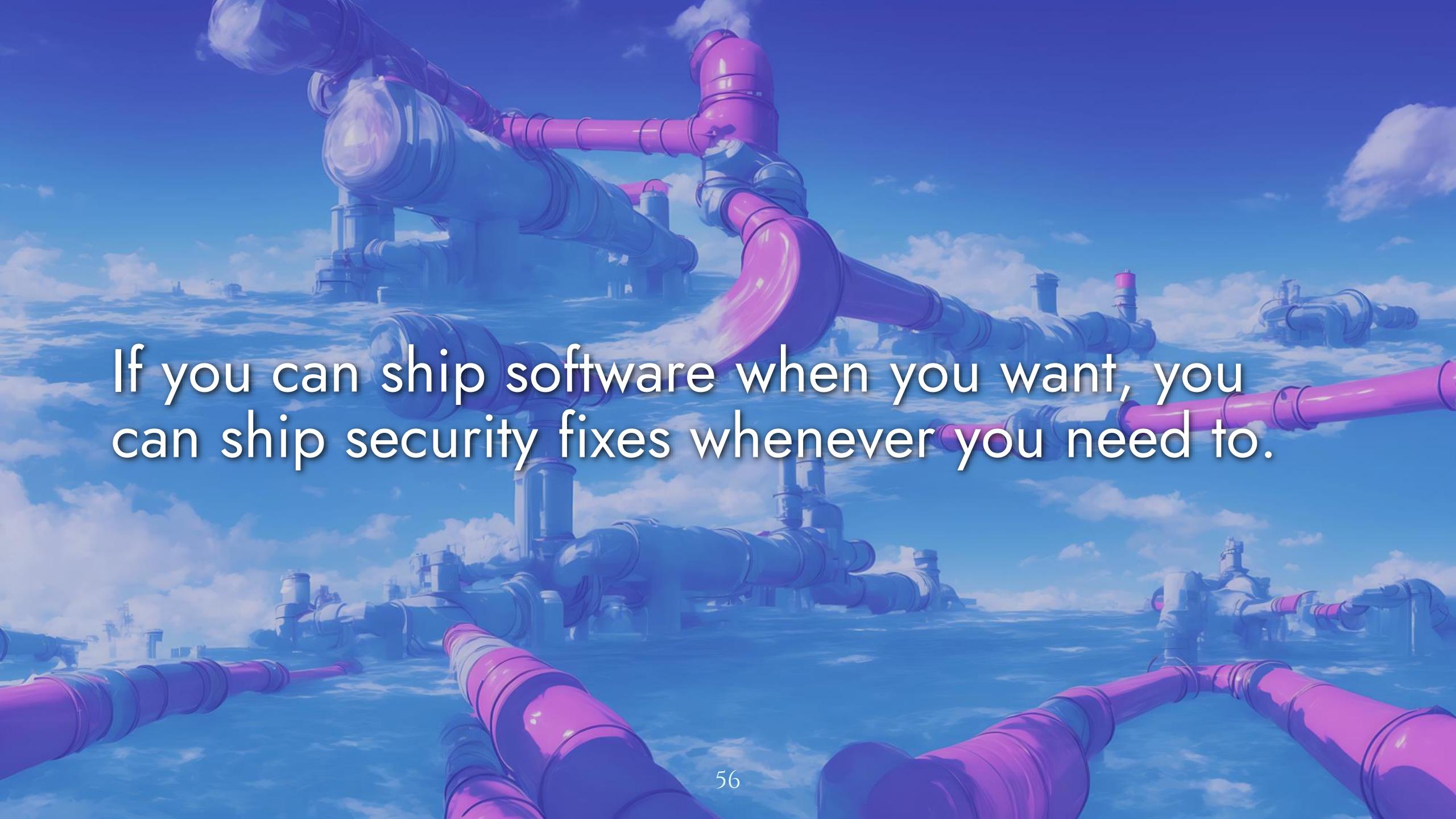
“Services must communicate over TLS and validate remote certificates.”

The background of the slide features a vibrant, abstract illustration of several glowing spheres, resembling celestial bodies or energy particles. These spheres are primarily composed of translucent blue and pink hues, with some showing internal swirling patterns. They are set against a dark, star-filled space with a nebula-like glow in the center.

“Only images built by our CI/CD system may run on the production Kubernetes cluster.”



“Secrets should be retrieved on demand from our secrets store instead of being baked into source code or deployment images.”

A futuristic industrial landscape set against a bright blue sky with scattered white clouds. The scene is filled with a complex network of large, glowing blue and pink pipes that appear to be floating in the air. These pipes are interconnected by various valves, fittings, and support structures. In the background, there are several floating platforms or small buildings, also connected by the pipe system. The overall aesthetic is clean, modern, and somewhat surreal.

If you can ship software when you want, you  
can ship security fixes whenever you need to.

Everything is recorded; you can set granular policy on who can deploy where and for what

The screenshot shows a web browser window with several tabs open at the top. The active tab is 'firewall-demo/banlist.txt' from the 'shortridge-sensemaking/firewall-demo' repository. The URL in the address bar is <https://github.com/shortridge-sensemaking/firewall-demo/blob/main/banlist.txt>. The GitHub interface includes a search bar, a star icon, and various navigation links like 'Code', 'Pull requests', 'Actions', 'Security', 'Insights', and 'Settings'. Below the header, the file 'banlist.txt' is displayed. The file content is a text-based banlist with comments and specific IP addresses listed. A commit history is visible above the code editor, showing a recent revert by 'swagitda'.

```
1  #
2  #
3  #
4  # Binary Defense Systems Artillery Threat Intelligence Feed and Banlist Feed
5  # https://www.binarydefense.com
6  #
7  # Note that this is for public use only.
8  # The ATIF feed may not be used for commercial resale or in products that are charging fees for such services.
9  # Use of these feeds for commerical (having others pay for a service) use is strictly prohibited.
10 #
11 #
12 #
13
14 1.10.241.225
15 1.85.49.110
16 1.183.12.102
17 1.206.27.29
18 1.215.138.43
19 1.235.198.19
```

CI/CD can help us with patching and keeping dependencies up to date

## Improper Input Validation in Octocat #120

Open

Opened on Aug 6 on octocat (pip) · requirements.txt



Bump Octocat from 8.0.8 to 8.0.9 ✓

Updating octocat in requirements.txt would resolve 4 Dependabot alerts



Review security update

Severity score

High

9.5 / 10

Dismiss as ▾

Show breakdown

### Vulnerable calls (Beta)

Give feedback

Showing the first of 4 calls to known vulnerable functions in octocat 8.0.8: `octocat.build` and `octocat.load`

fresh-co/monalisa/main.py

4 calls in fresh-co/monalisa/main.py

```
209     mona_data = {}
210     octocat_location = self.env.get_template(filepath)
211     octocat = octocat.build(octocat_location, Loader=octocat.load)
```

Vulnerable function called: `octocat.build`

```
212
213     mona_data["monalisa"] = octocat
```

Weakness types

CWE-20

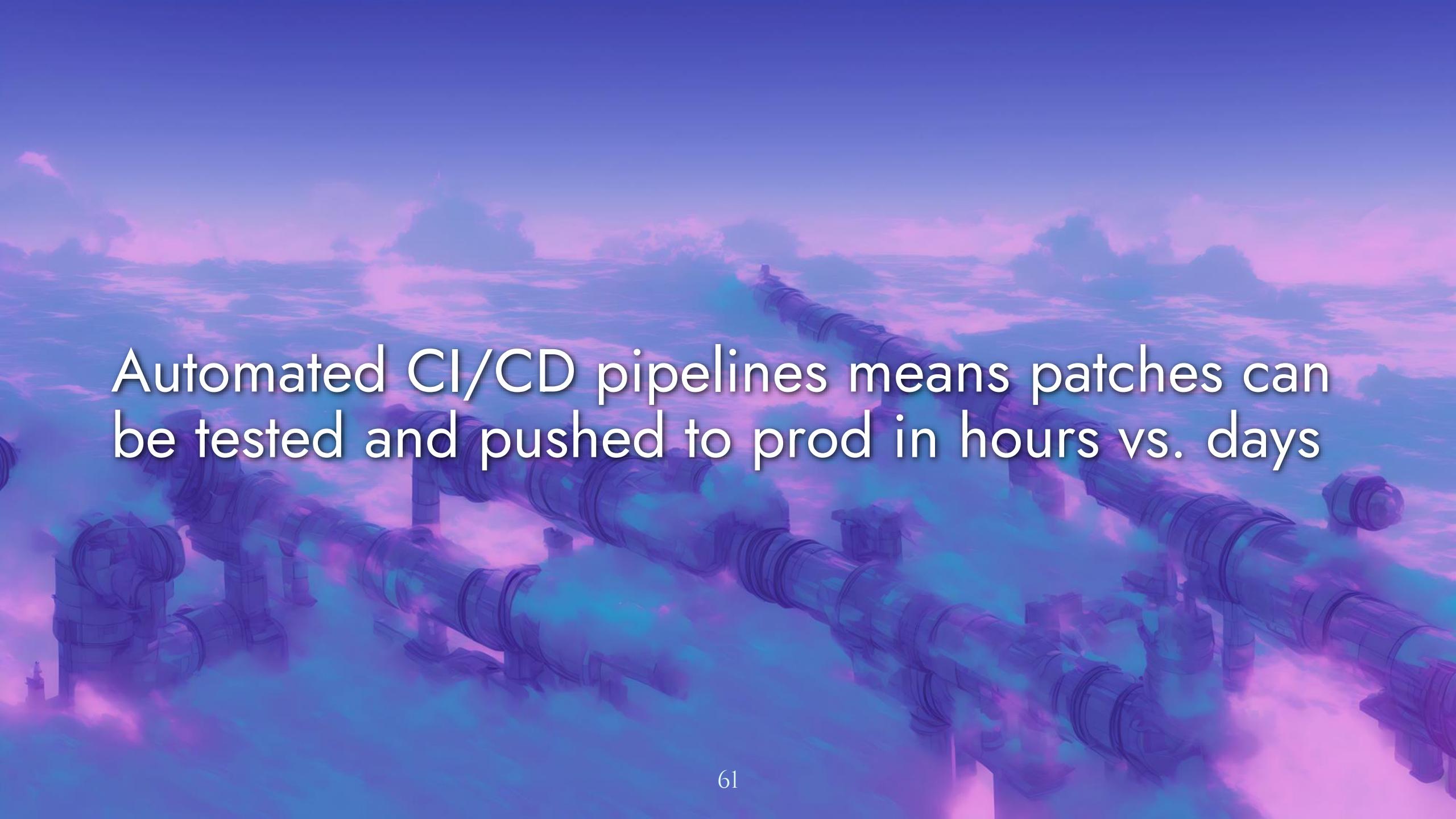
CVE ID

CVE-2020-1747

GHSA ID

GHSA-6757-jp84-gxfx

See all your affected repositories

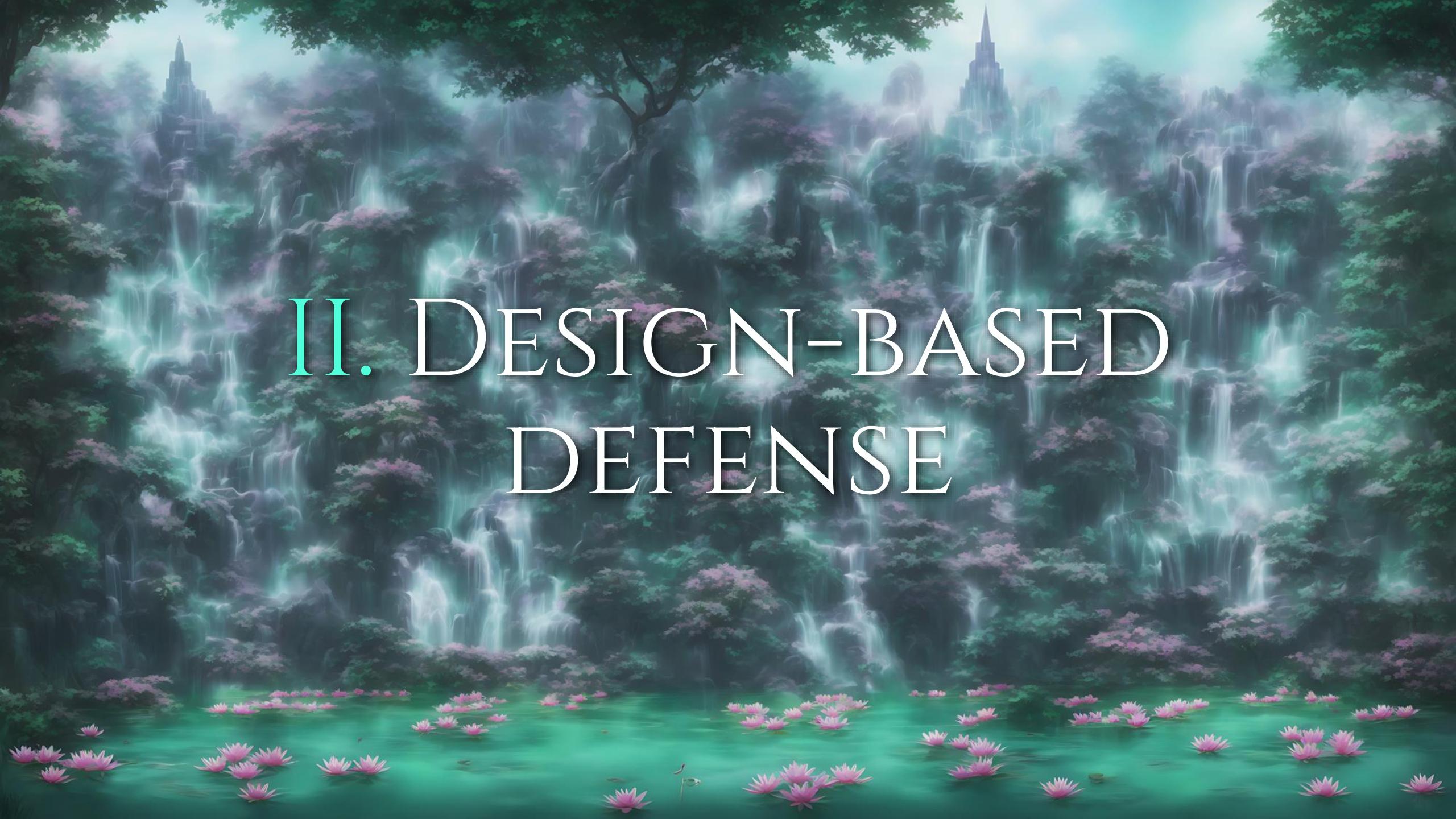
A large network of pipes, some transparent and glowing with a blue light, against a background of a cloudy sky with a purple and blue hue.

Automated CI/CD pipelines means patches can  
be tested and pushed to prod in hours vs. days

Update-and-patch cycles become an automatic, daily affair, freeing time for other priorities

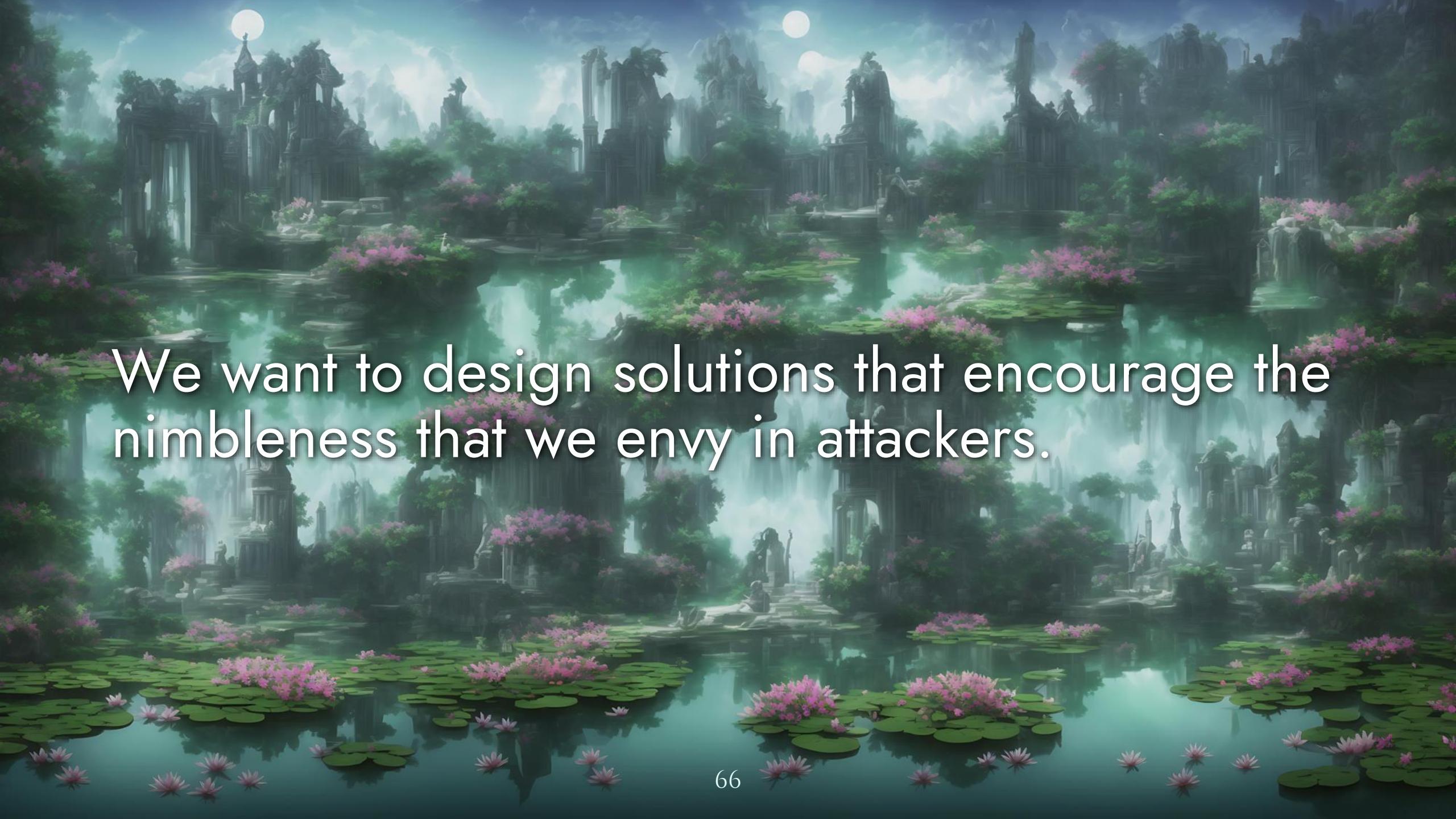
The background features a futuristic cityscape with a dense grid of skyscrapers. A prominent feature is a large, glowing blue pipe system that stretches across the frame, from the foreground to the horizon. The sky above the city is a deep, hazy blue, suggesting a futuristic or alien environment.

tl;dr CI/CD lets us move faster and track the things we do – or revert (attackers can't do so)

The background of the slide is a dense, vibrant green forest. Large, multi-tiered waterfalls cascade down the rocky cliffs, their paths illuminated by a soft, glowing light that highlights the mist and spray. The forest floor is carpeted with numerous pink lotus flowers and other small, delicate pink blossoms. In the upper left and right corners, large, leafy trees with dark branches frame the scene. The overall atmosphere is one of a serene, magical, and untouched natural environment.

## II. DESIGN-BASED DEFENSE

How should we prioritize the types of solutions we design? Are some better than others?



We want to design solutions that encourage the nimbleness that we envy in attackers.

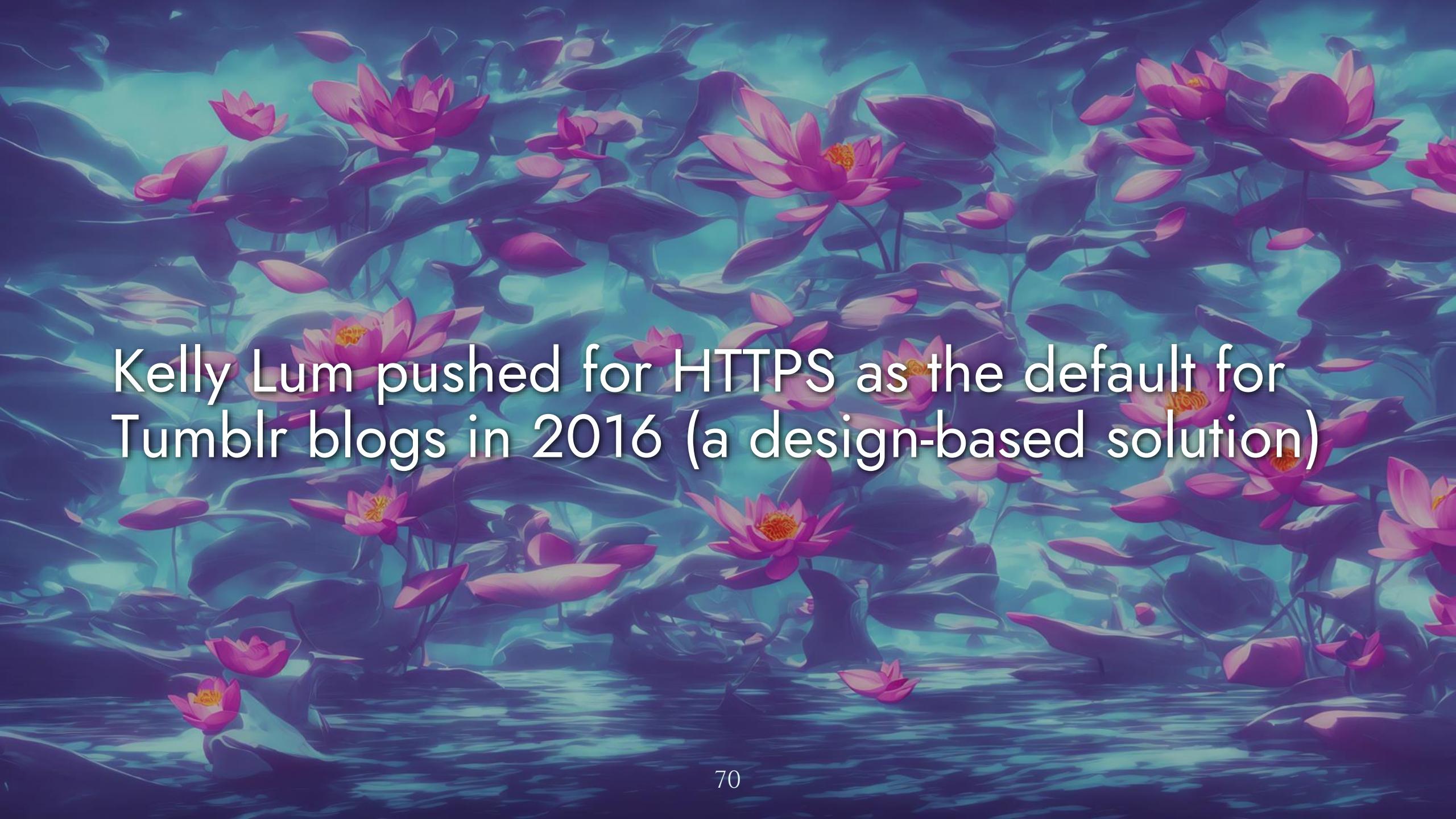


*Figure 7-3. The Ice Cream Cone Hierarchy of Security Solutions*

The background features a vast, turbulent sky filled with dark, billowing clouds. Interspersed among the clouds are bright, glowing patches of light in shades of cyan, magenta, and white, resembling distant lightning strikes or perhaps the reflection of a setting sun. In the lower portion of the image, a dark, choppy body of water reflects the colors of the sky. The overall atmosphere is one of a powerful, dynamic natural scene.

“Human fallibility is like gravity, weather, and terrain, just another foreseeable hazard.”

Finite cognitive resources; competing pressures; exhaustion, stress, distraction...

A dense field of pink lotus flowers with large green leaves, floating on water.

Kelly Lum pushed for HTTPS as the default for  
Tumblr blogs in 2016 (a design-based solution)

Isolation, standardization, message buses,  
declarative dependencies, queues, failover...

A futuristic landscape featuring a vast, rolling terrain covered in glowing, organic-looking terraced fields. The fields emit a vibrant blue and green light, creating a sense of depth and depth. In the background, several large, luminous spheres of varying sizes float in the sky, casting a soft glow over the scene. The overall atmosphere is dreamlike and otherworldly.

MODULARITY

Modularity: allows structurally or functionally distinct parts to retain autonomy during periods of stress & allows for easier recovery from loss

A fantastical landscape featuring numerous floating rock islands of various sizes. Each island is densely covered in vibrant, fuzzy vegetation, primarily a bright pink or magenta color, resembling moss or a specific type of flower. The islands are interconnected by a network of thin, winding paths and small bridges. In the background, a large, luminous full moon hangs in a dark, cloudy sky, casting a soft glow over the scene. The overall atmosphere is dreamlike and otherworldly.

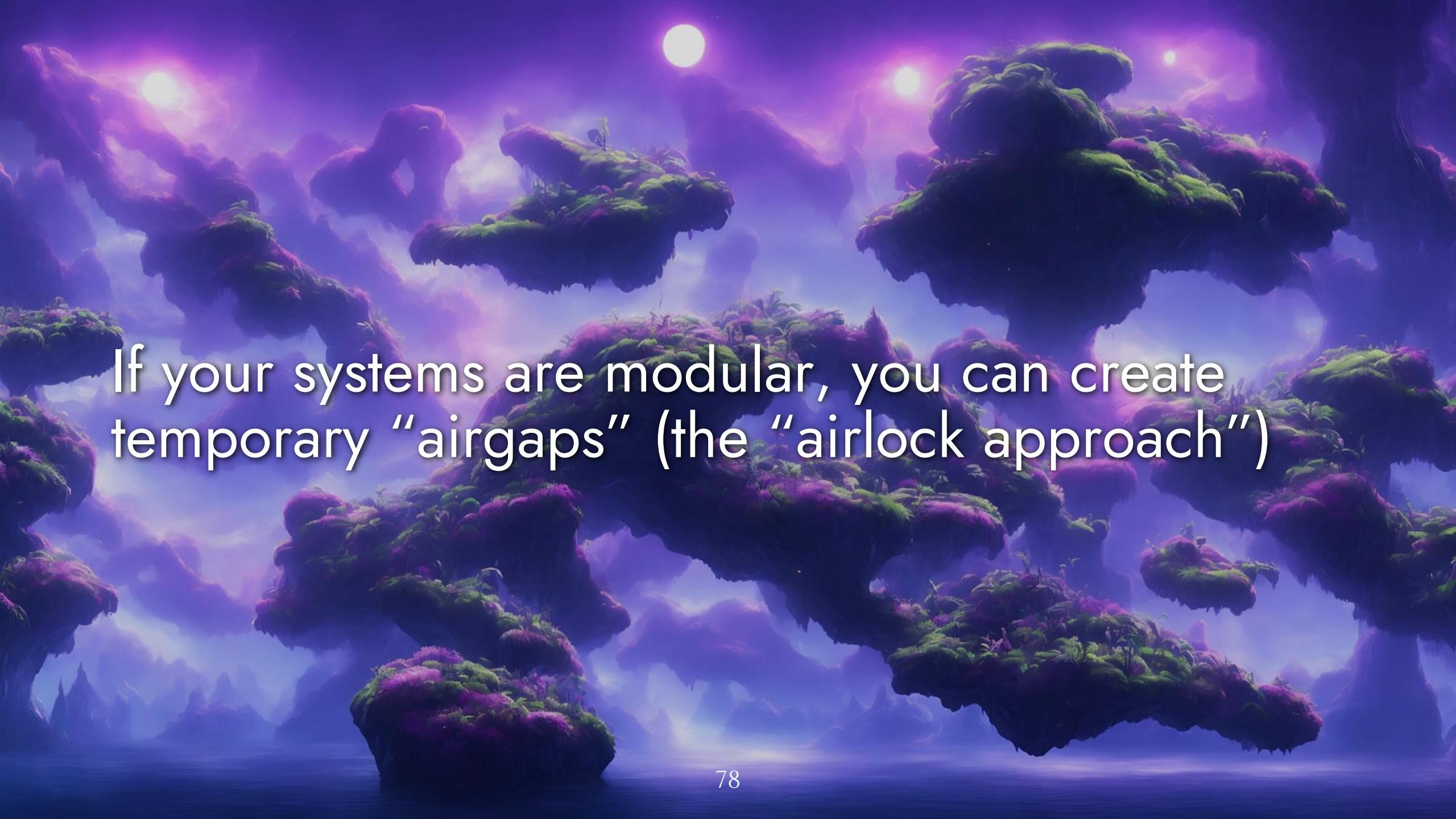
Unless components can fail independently, you  
don't have modularity in the resilience sense.

Queues and message brokers support modularity, each in different ways...

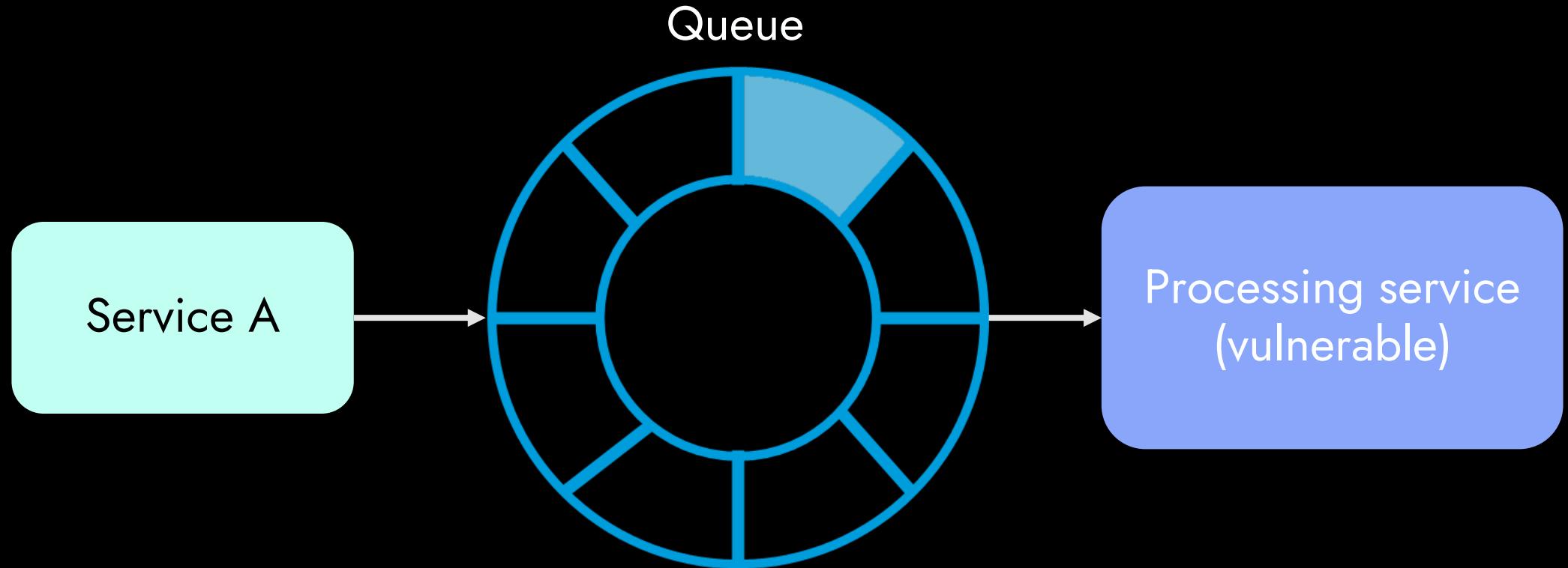


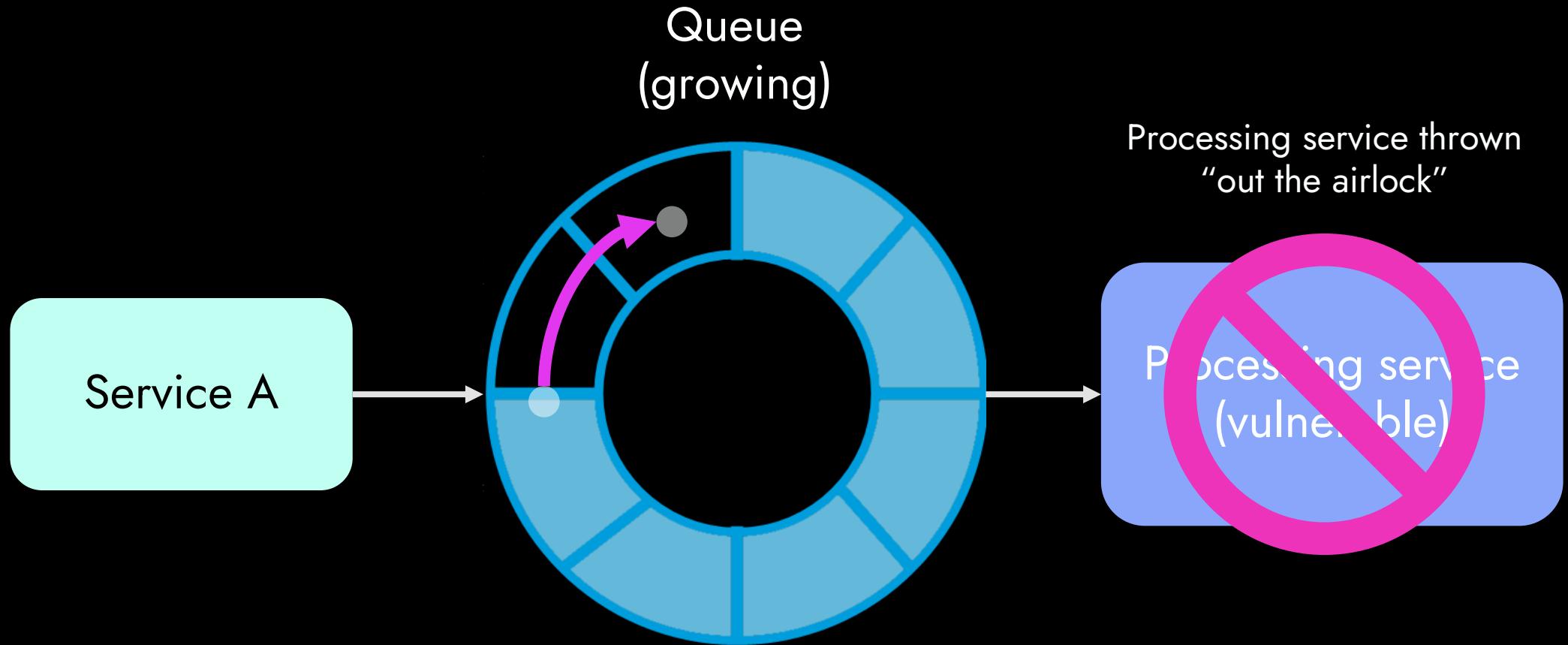
A queue adds a buffer; a message broker can replay and make return code non-blocking.

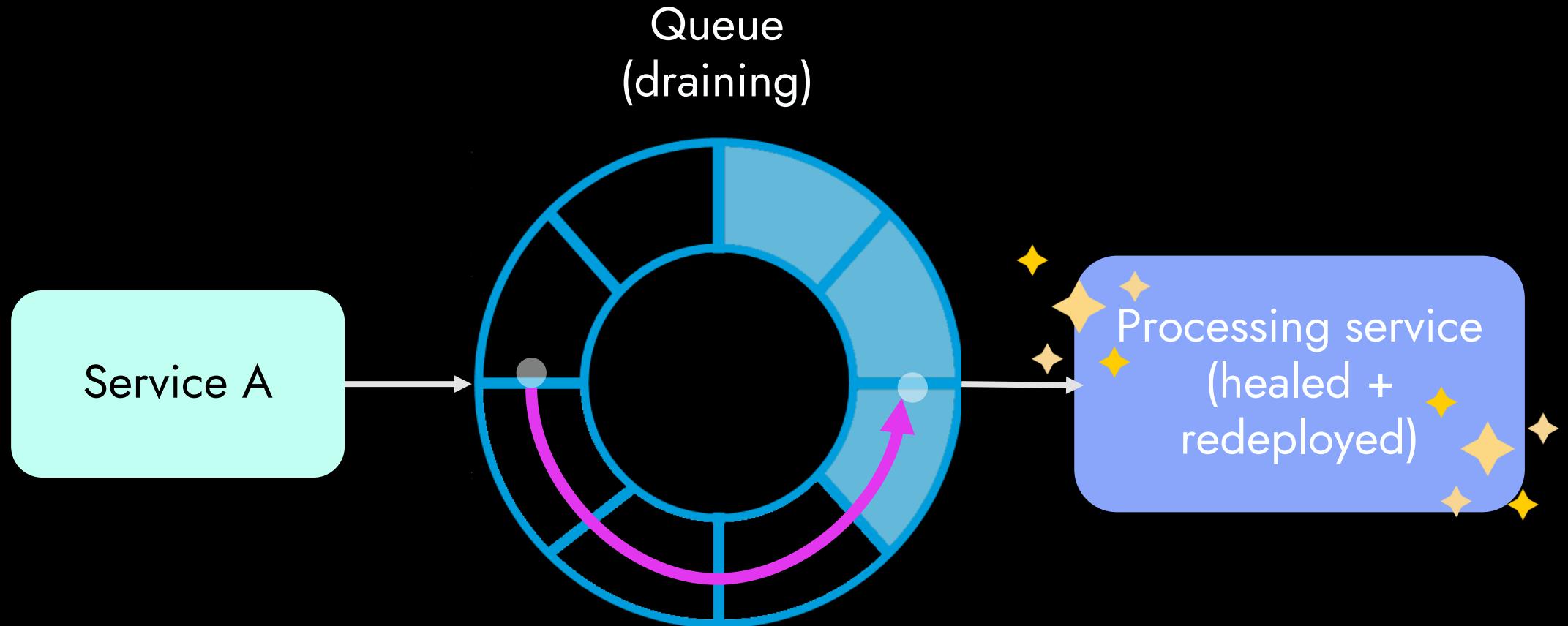
Both tools standardize how services pass data around and provide a centralized view.



If your systems are modular, you can create temporary “airgaps” (the “airlock approach”)







The background is a vibrant, futuristic landscape featuring several floating, spire-topped cities against a dark blue sky. These cities are surrounded by large, billowing clouds that glow with a bright cyan light from within. The overall atmosphere is mysterious and otherworldly.

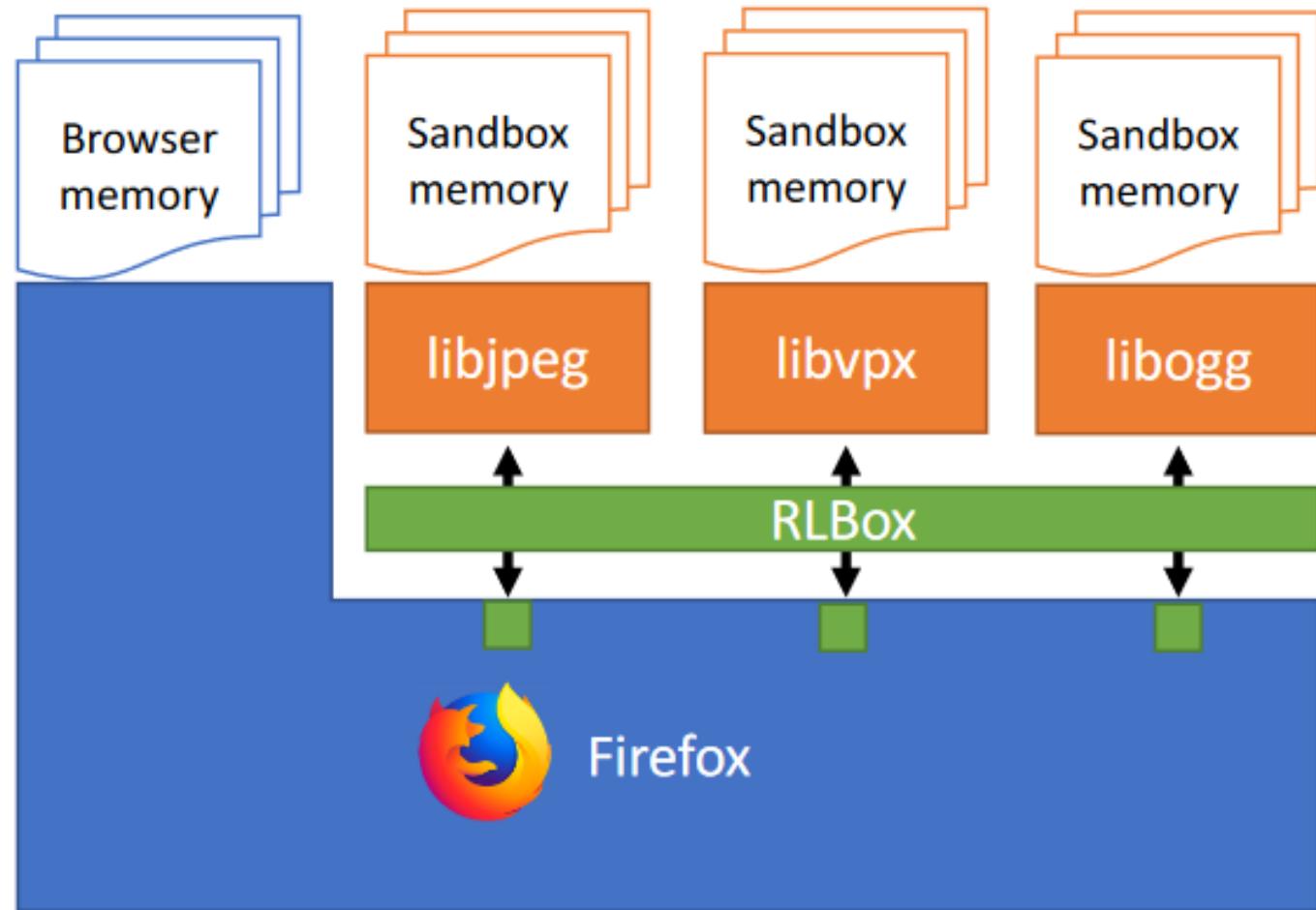
Modularity minimizes incident impact – think  
ransomware in serverless (it doesn't happen)

Modularity allows for basic encapsulation and separation of concerns... and supports isolation



Here's what it's like to live in 2023 with a strong engineering culture:

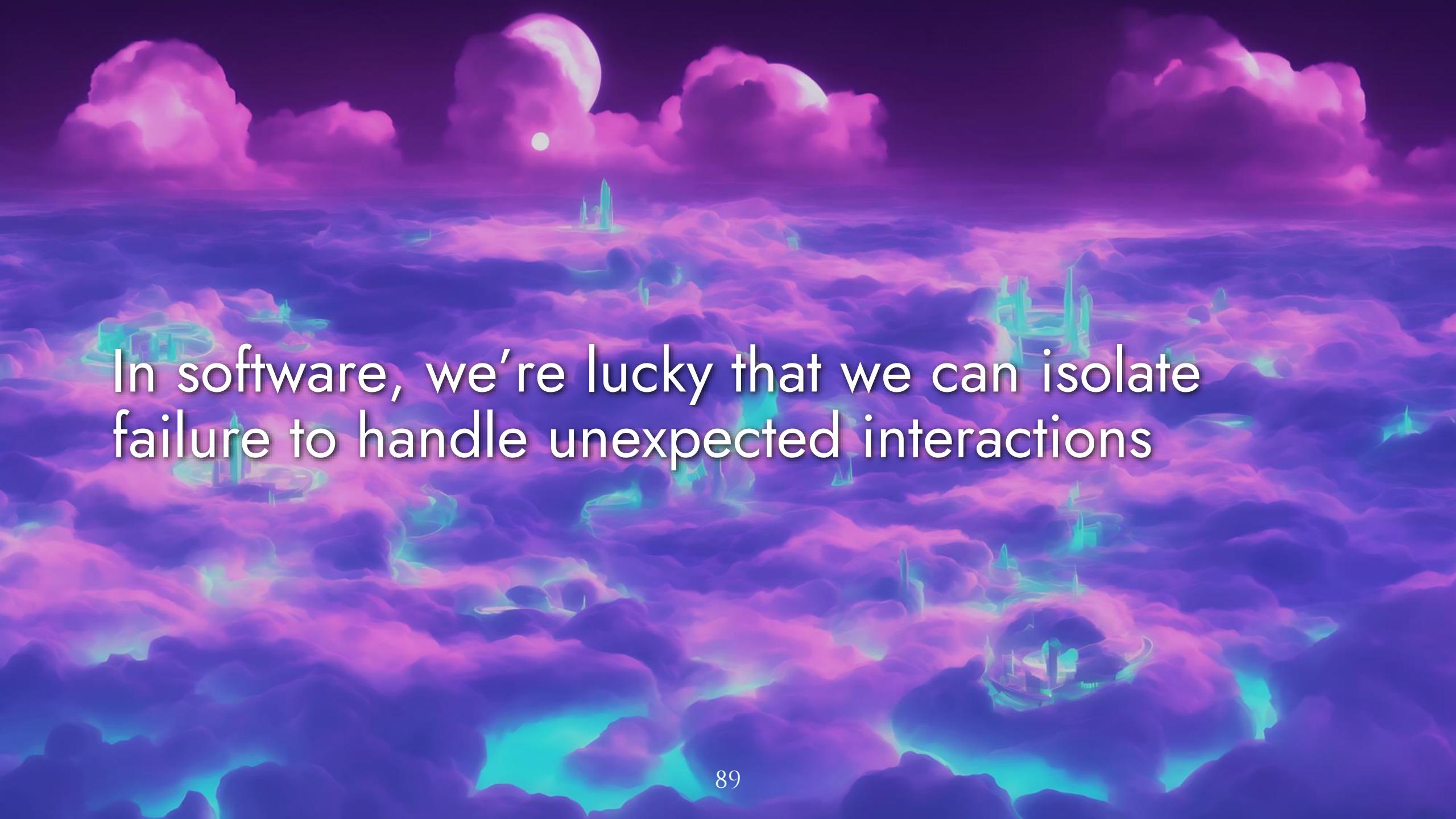
# RLBox: trap C code in a WebAssembly (Wasm) sandbox to isolate hazardous subcomponents



A whimsical landscape featuring several white castles with tall spires floating amidst large, fluffy clouds against a dark blue sky. Two vibrant rainbows arc across the scene, one in the upper left and another in the upper right. The overall atmosphere is dreamlike and fantastical.

Imagine not worrying about Oday anymore\*.

You've been so focused on AI you've missed groundbreaking things like this. It's sad. :(



In software, we're lucky that we can isolate failure to handle unexpected interactions

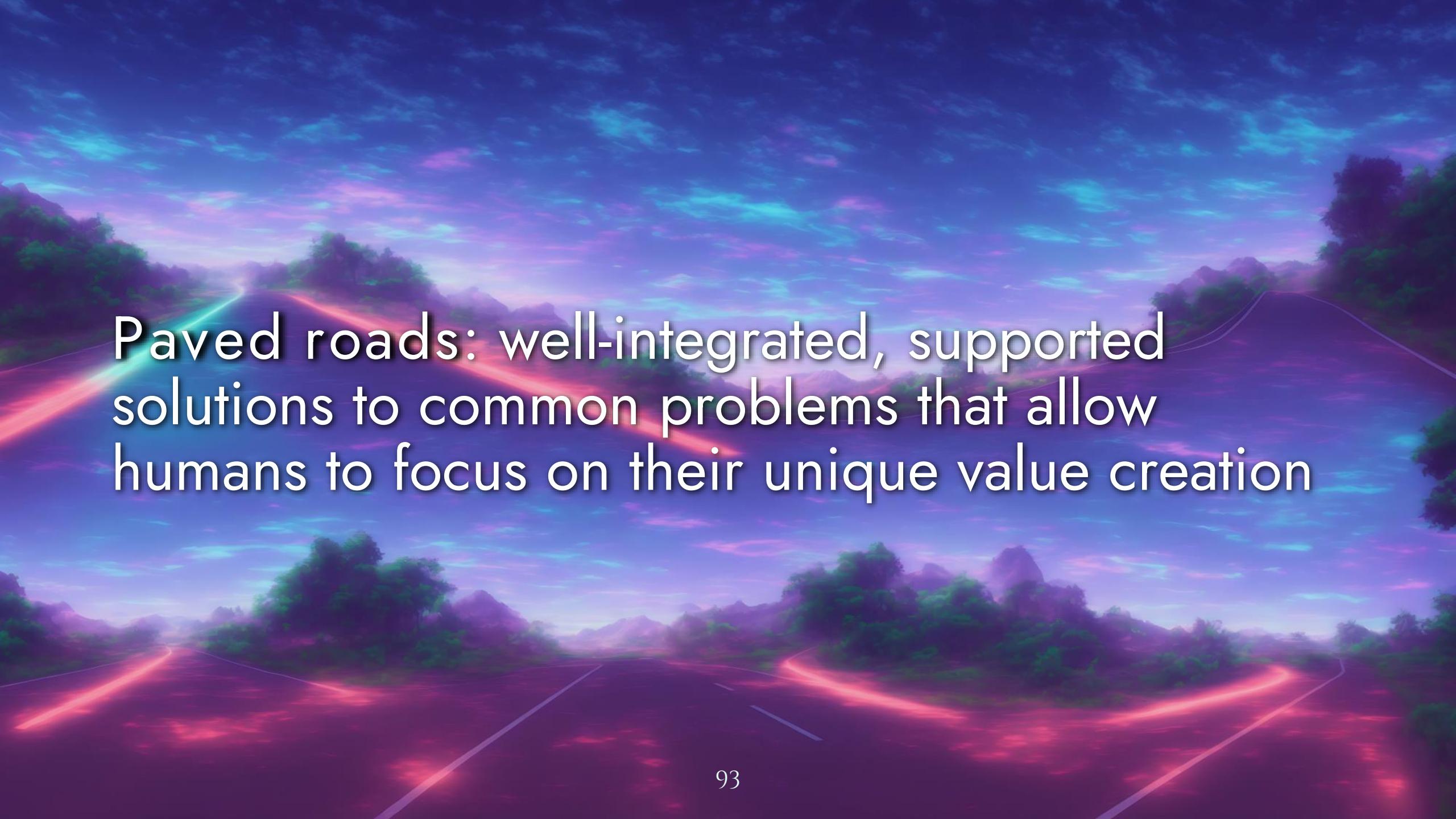
Start “boring”: set AWS security groups – or  
use serverless functions, containers, or VMs



If a vulnerable component is in a sandbox, the attacker faces a challenge to reach their goal

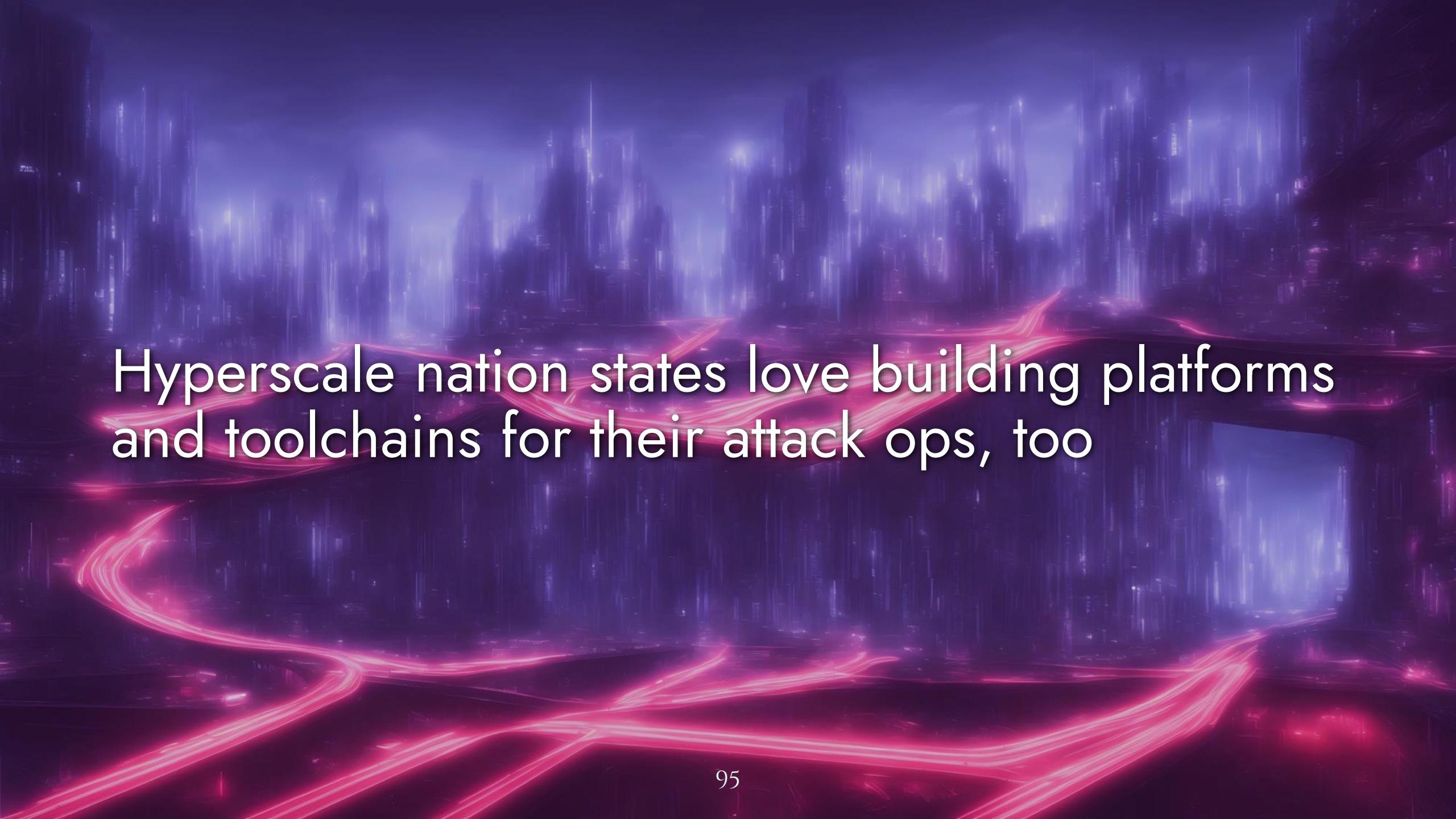
An aerial photograph of a multi-level highway interchange at night. The roads are illuminated by bright, glowing light trails from moving vehicles, creating a vibrant, colorful pattern against the dark asphalt. The interchange is set against a backdrop of green hills and a sky filled with soft, pastel-colored clouds. The overall scene is dynamic and emphasizes the movement and infrastructure of modern transportation.

PAVED ROADS

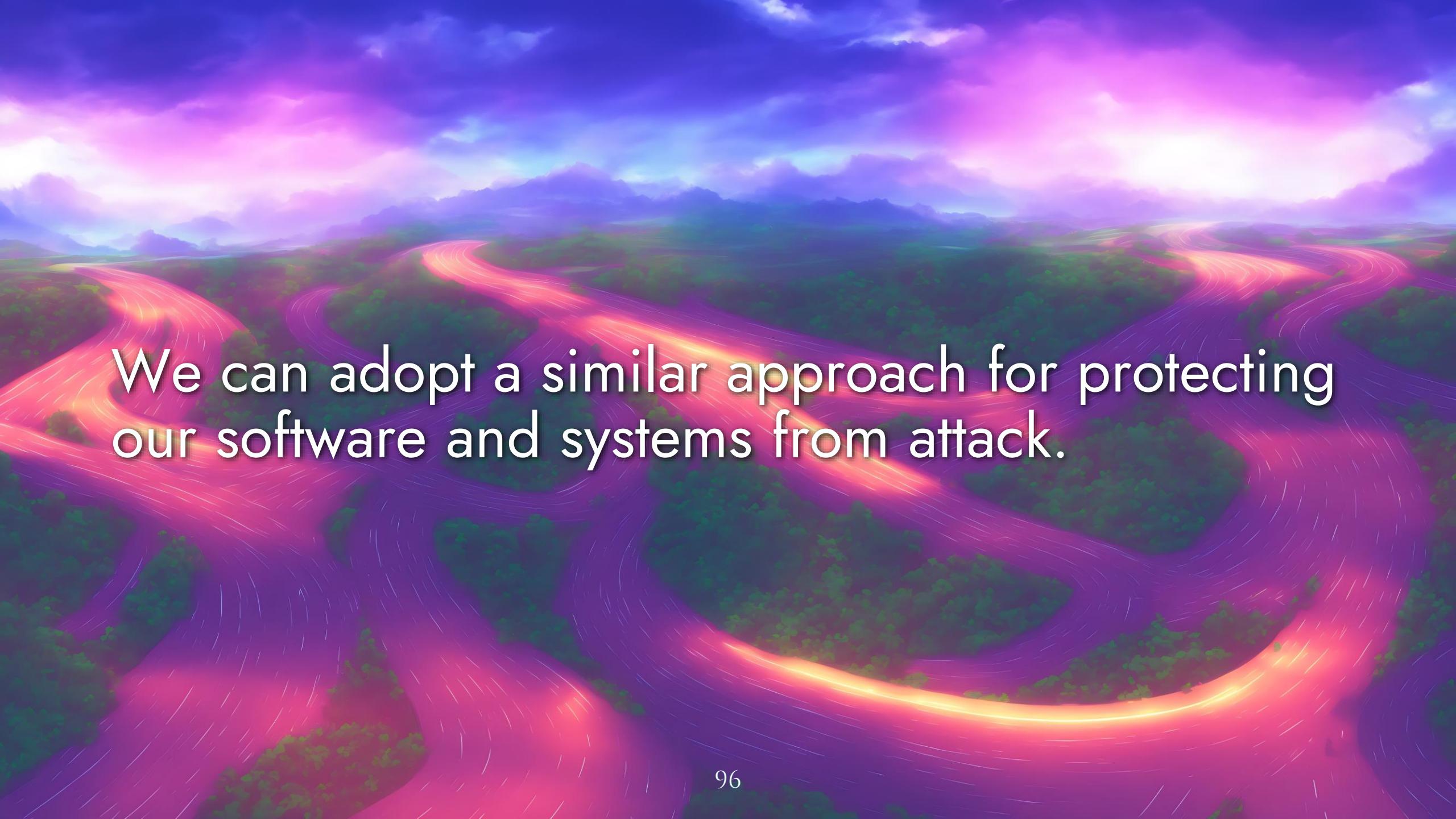
A wide-angle photograph of a paved road curving through a dense forest. The sky is filled with dramatic, colorful clouds in shades of blue, purple, pink, and orange, suggesting a sunset or sunrise. The road's surface reflects the warm light of the sky. The surrounding trees are dark green and silhouetted against the bright sky.

Paved roads: well-integrated, supported  
solutions to common problems that allow  
humans to focus on their unique value creation

Attackers have paved roads, like Cobalt Strike  
– it makes the easy way the pwnful way.

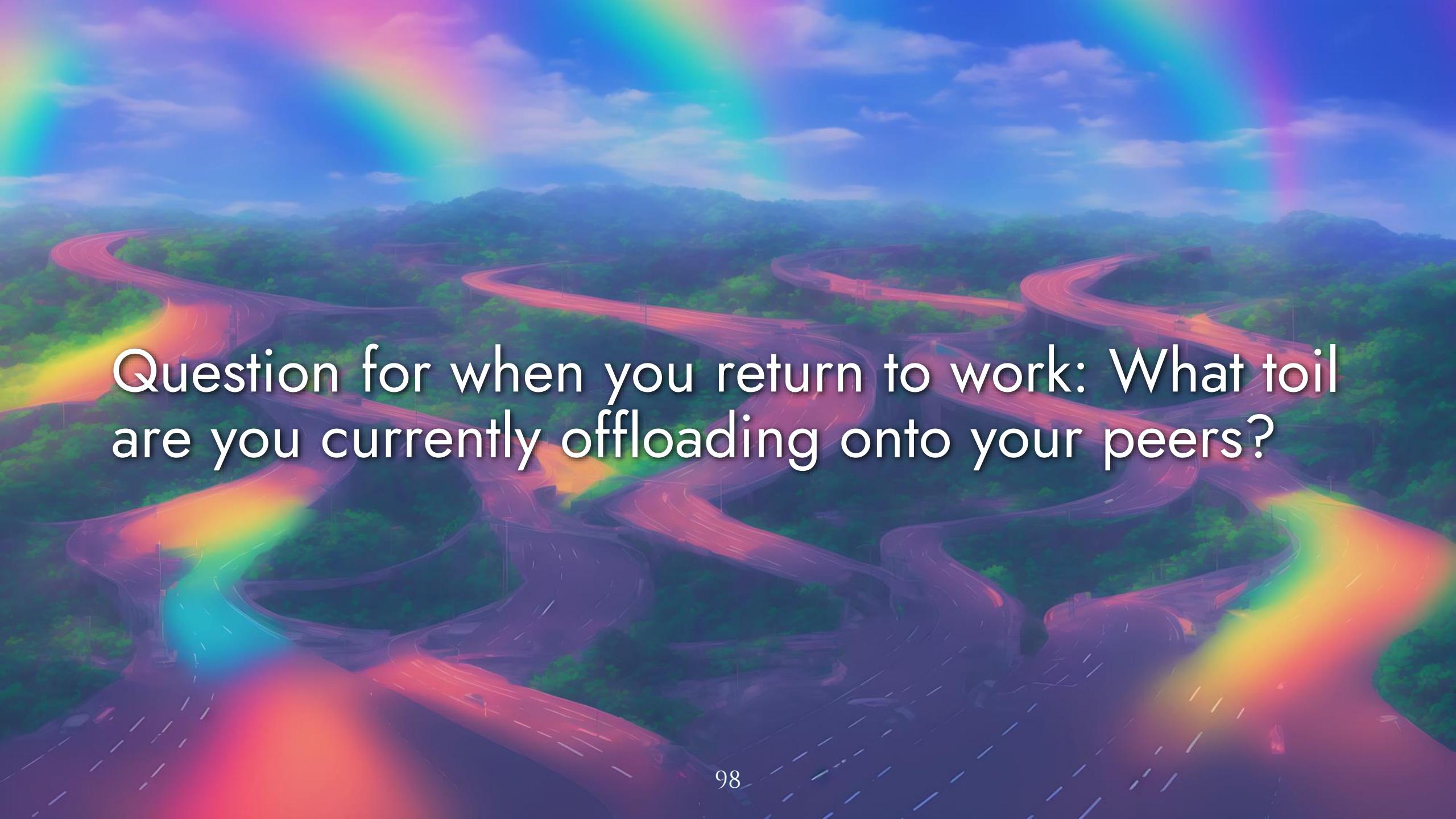


Hyperscale nation states love building platforms  
and toolchains for their attack ops, too

The background of the slide features a vibrant, abstract landscape. It consists of a series of winding, glowing red and orange paths or roads that snake through a dark green, forested terrain. In the distance, there are faint, hazy outlines of mountains under a sky filled with soft, pastel-colored clouds in shades of pink, purple, and blue.

We can adopt a similar approach for protecting our software and systems from attack.

Netflix: Wall-E framework turns security requirements into filters to replace checklists

The background of the slide is a photograph of a complex highway interchange with multiple levels of overpasses and ramps winding through a lush green forested hillside under a clear blue sky.

Question for when you return to work: What toil  
are you currently offloading onto your peers?

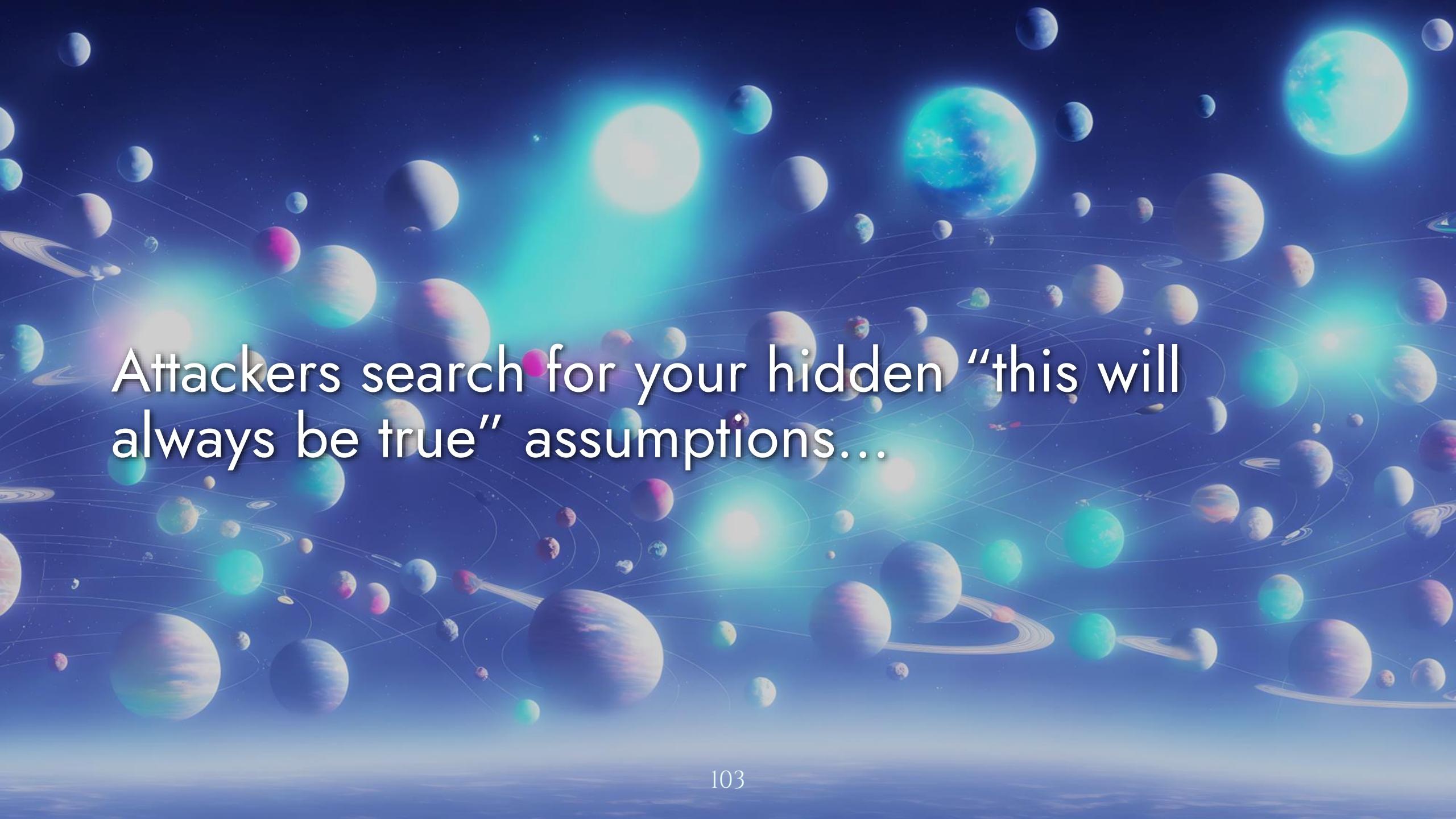
“The bulk of the ‘going internet-facing’ checklist boiled down to one item: Will you use Wall-E?”

**Block:** enabling backend services to securely connect across business unit boundaries

The background features a futuristic landscape with a blue and purple color palette. It consists of a large, semi-transparent grid overlaid on a dark blue base. Numerous glowing, translucent spheres of various sizes are scattered across the grid, some with internal patterns. The overall effect is a complex, interconnected system, possibly representing a network or a simulation environment.

### III. SYSTEMS THINKING

Attackers think in systems while defenders think in components. It doesn't have to be this way.

The background of the slide is a vibrant, multi-layered illustration of space. It features numerous planets of various sizes, colors, and textures, ranging from small blue spheres to large, detailed worlds with continents and clouds. Some planets have prominent rings, resembling Saturn or Uranus. The space is filled with glowing nebulae, some appearing as soft, translucent clouds and others as bright, intense points of light. A few small, distant stars are scattered across the dark blue void. The overall effect is a sense of depth and the vastness of the universe.

Attackers search for your hidden “this will  
always be true” assumptions...

Then they ask, “you say this will always be true; is that the case?” to break those assumptions

Attackers target our “this will always be true” assumptions that exist all over our stack.

PARSING THIS STRING  
WILL ALWAYS BE FAST



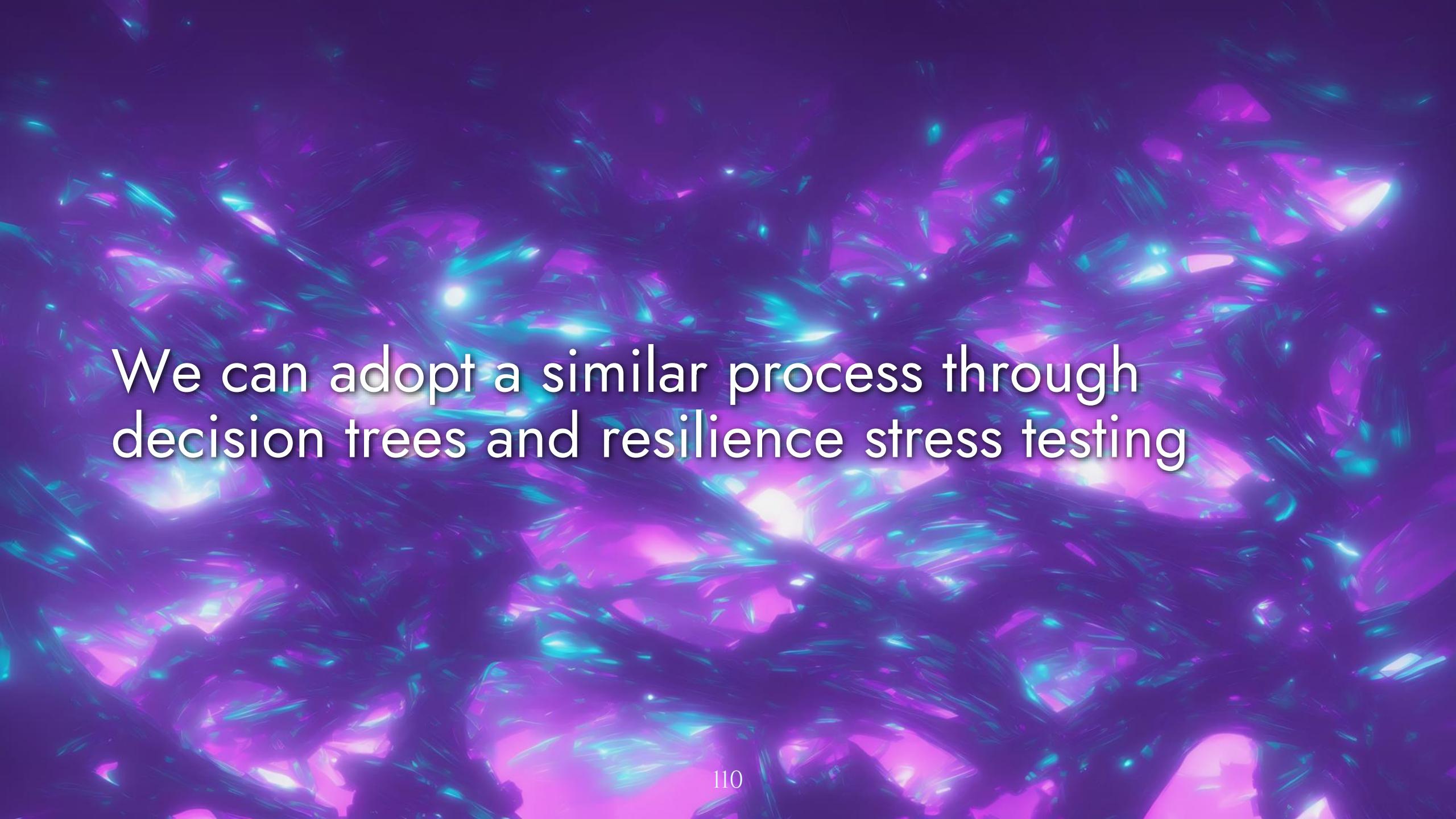


MESSAGES ON THIS  
PORT WILL ALWAYS  
BE POST-AUTH

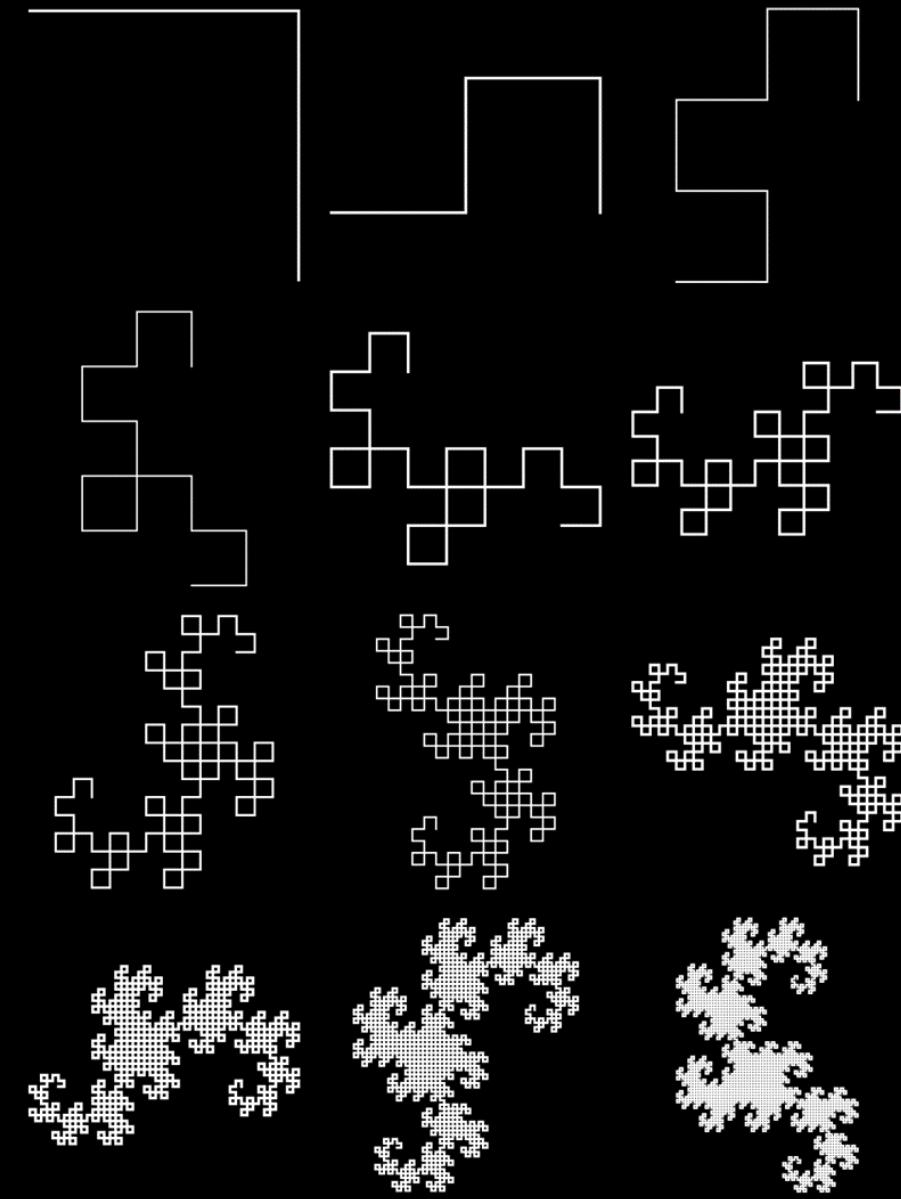


AN ALERT WILL ALWAYS  
FIRE IF A MALICIOUS  
EXECUTABLE APPEARS

The attacker thinks, “They say X here, but I can show that it isn’t quite true... interesting. Let’s keep looking to see if they’re just a little wrong or *really* wrong.”



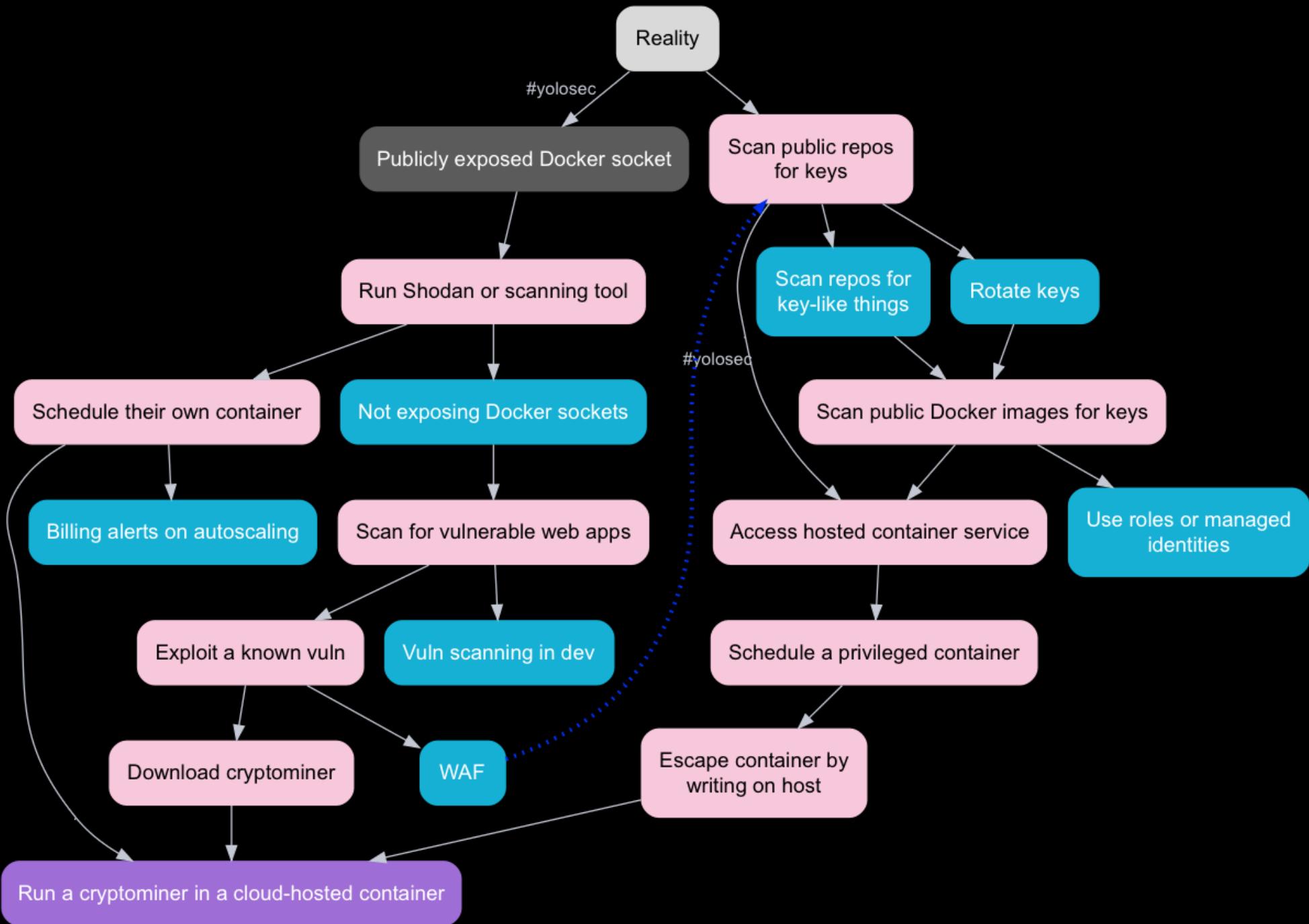
We can adopt a similar process through decision trees and resilience stress testing



The background of the slide features a complex, abstract fractal pattern. It consists of numerous thin, glowing blue lines that form intricate, organic shapes resembling leaves or petals. These lines are set against a dark purple background and are illuminated from within, creating a glowing effect. The overall aesthetic is futuristic and organic.

We can refine our mental models continuously  
rather than waiting for attackers to exploit them

### (Example) Attack Tree for Cryptominer in a Container



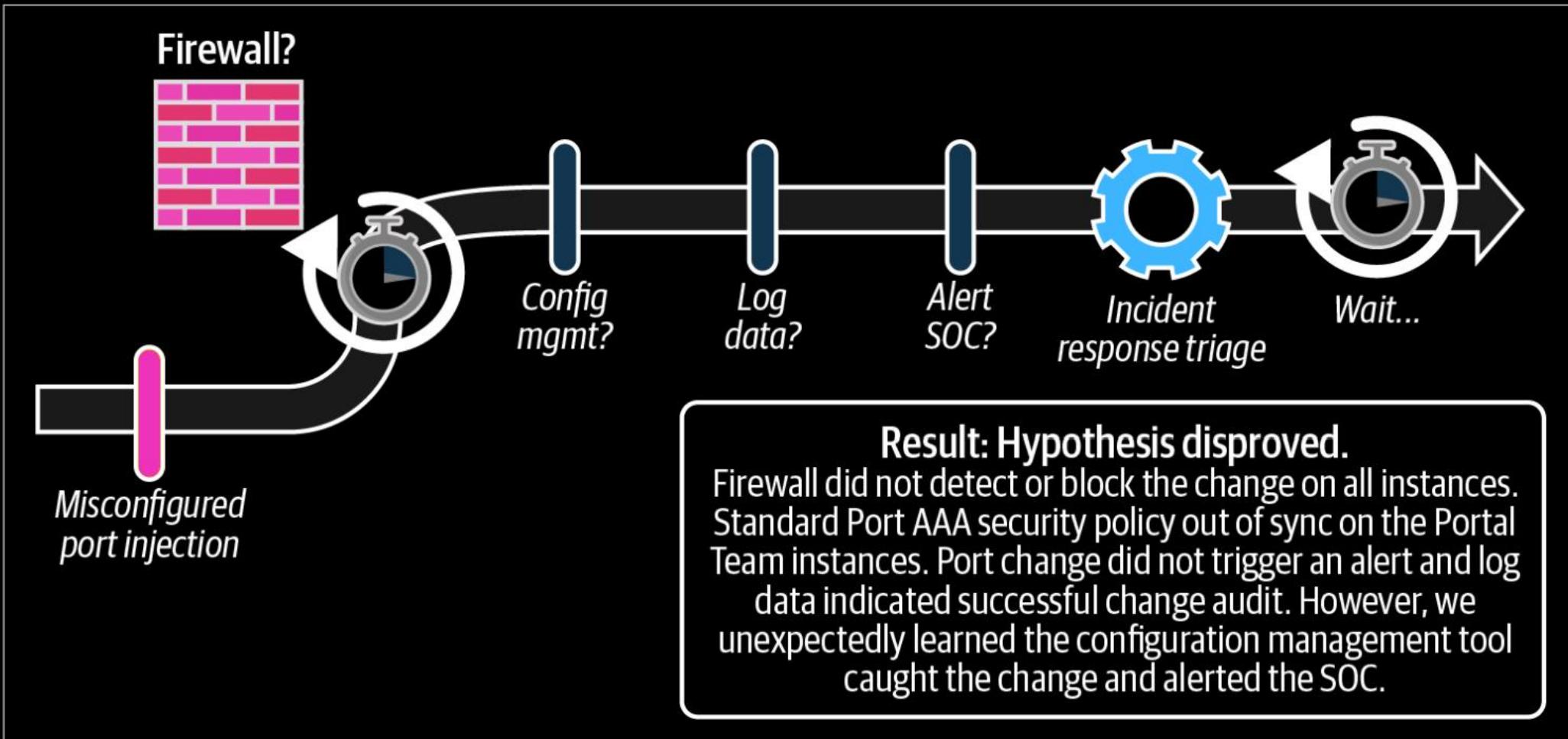


Resilience stress tests help us identify the confluence of conditions where failure happens

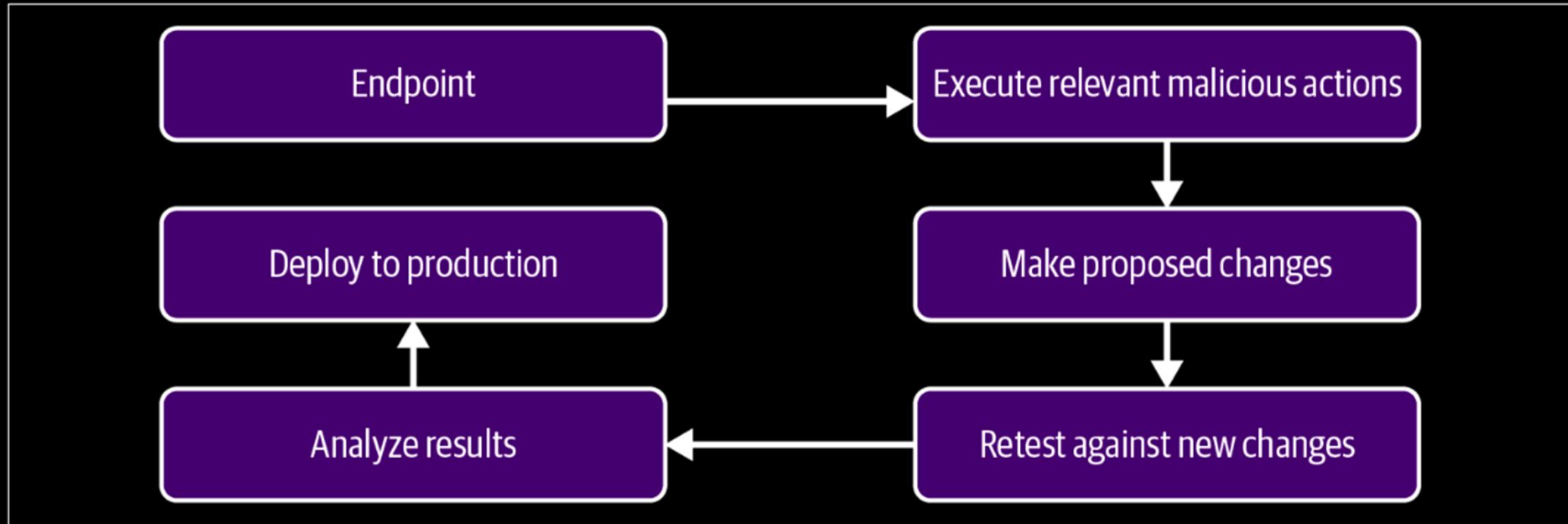
How do disruptions impact the entire system's ability to recover and adapt?



We can move fast and observe how failure unfolds in our systems through experiments

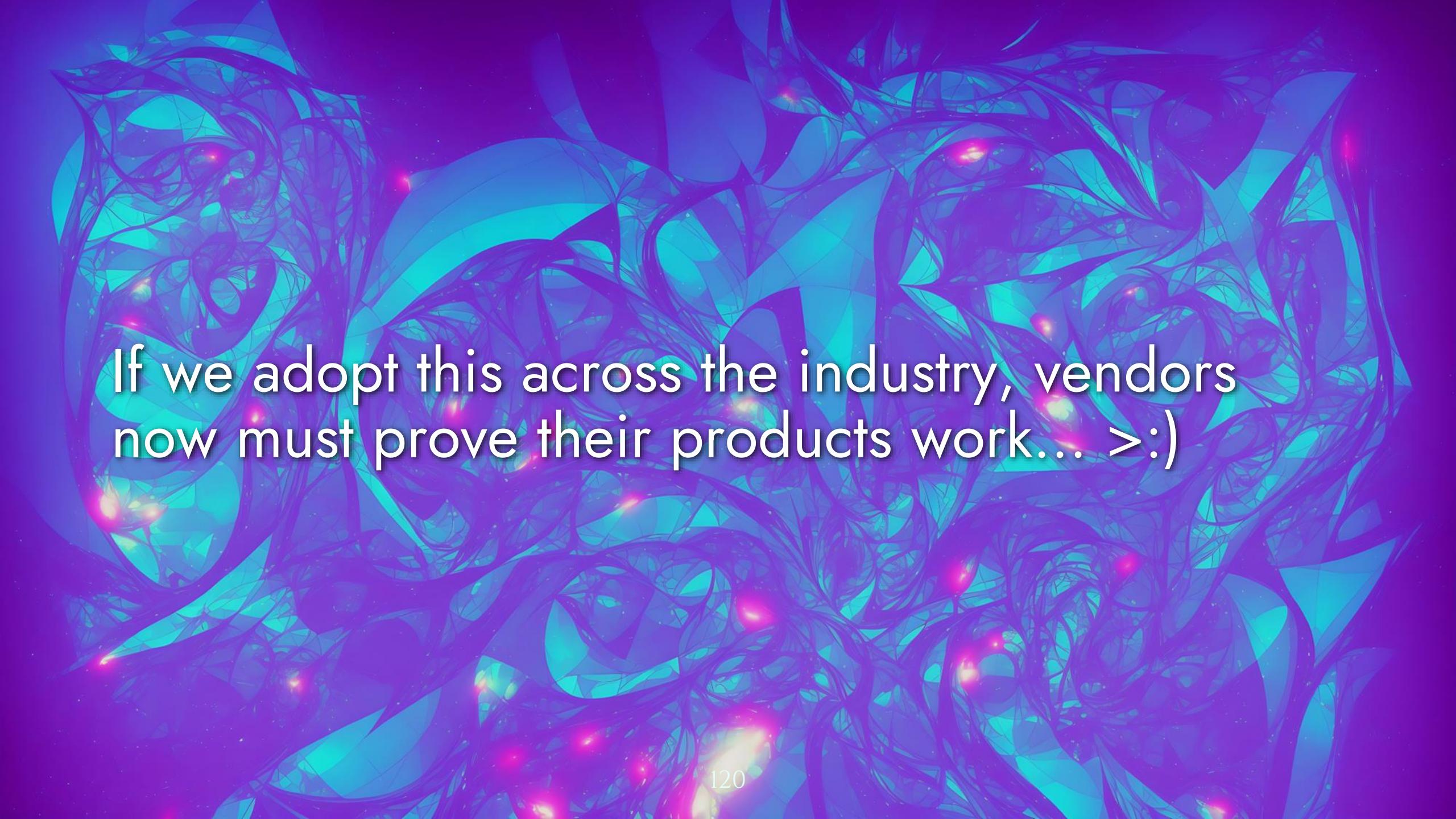


*Figure 2-6. An example security chaos experiment simulating a misconfigured port injection scenario*



*Figure 9-5. Engineering workflow change evaluation*

Verizon: deploy a pod containing known vulns  
on a target cluster to test security controls

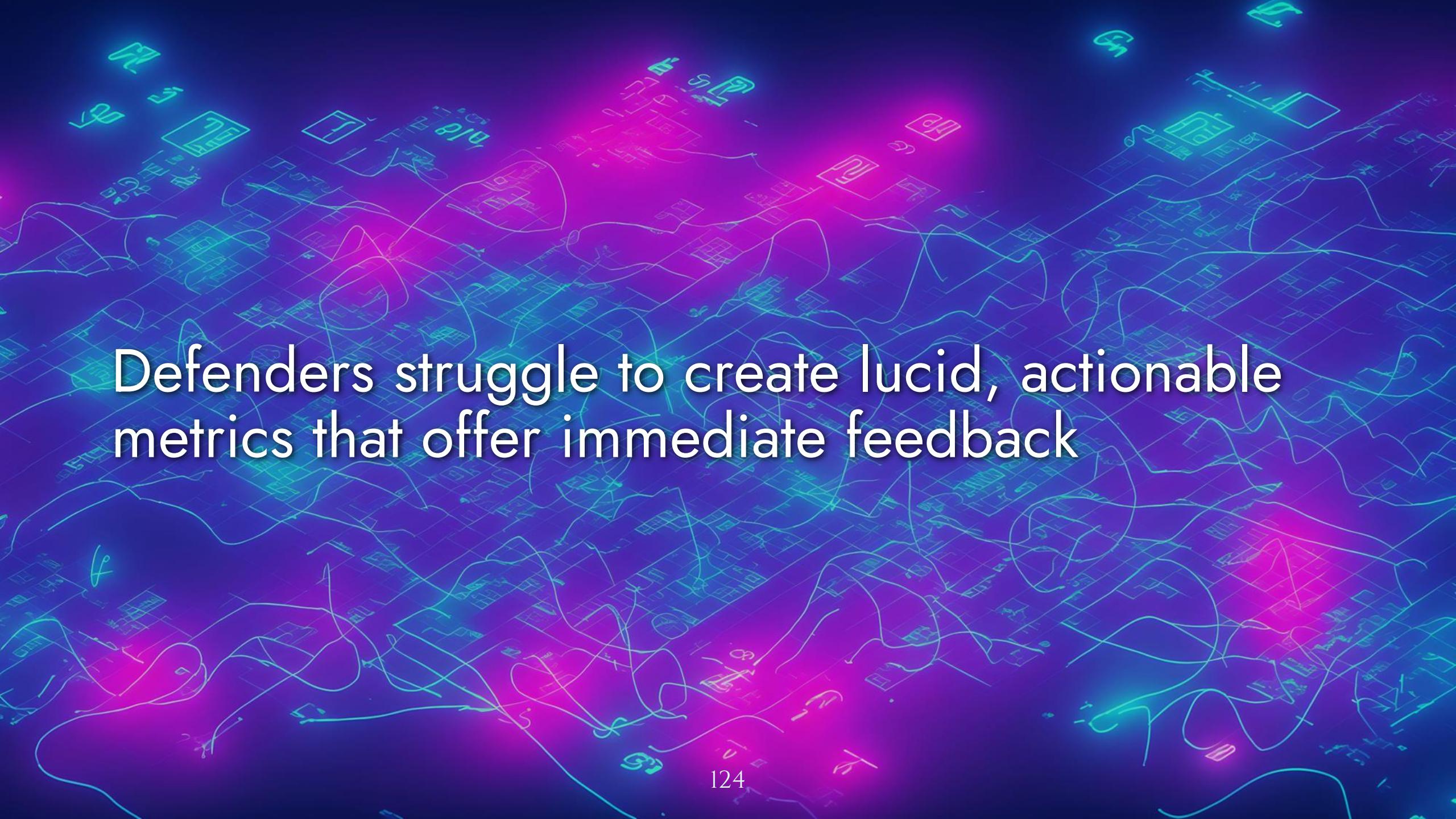


If we adopt this across the industry, vendors  
now must prove their products work... >:)

## IV. TANGIBLE SUCCESS

Attackers can measure tangible success and receive immediate feedback on their metrics

Do they have access, how much access do they have, and have they accomplished their goals?

The background features a dense, glowing cityscape composed of numerous small, glowing green and blue rectangles representing data points or buildings. These points are interconnected by a complex web of thin, glowing lines forming a grid-like network. The overall effect is one of a highly advanced, digital urban environment.

Defenders struggle to create lucid, actionable metrics that offer immediate feedback

CISOs, your “risk coverage” and “time to detect” mean nothing, it’s embarrassing

# SYSTEM SIGNALS

Reliability signals also benefit systems security



Who deployed what and when?  
(like orchestrator and deployment logs)

# Who accessed what and when? (like cloud audit data)



Database logs, billing records, netflow,  
production crash dumps, error messages...

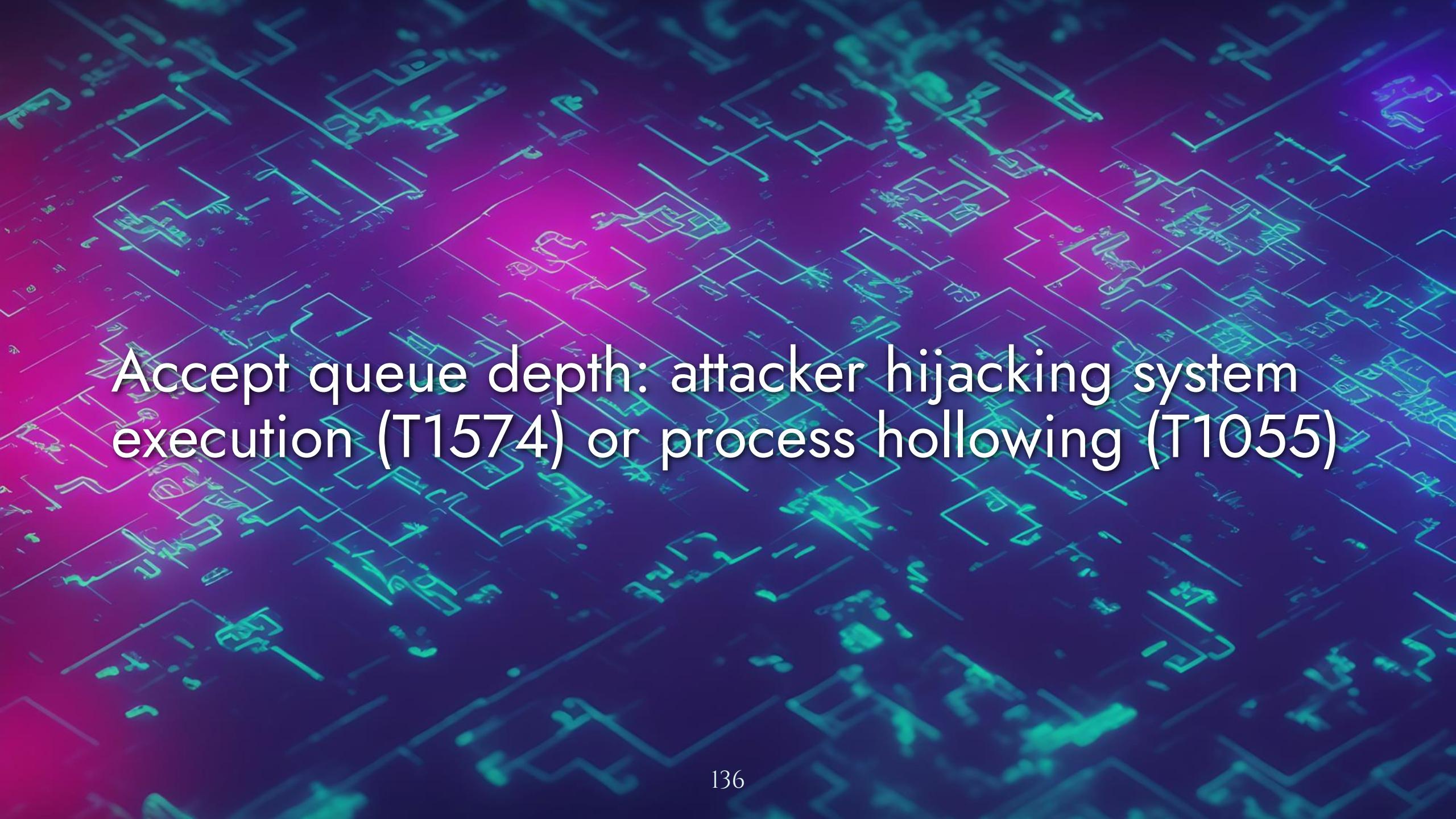
Traditional infosec doesn't measure load,  
latency, performance, or throughput (a mistake)

e.g. high CPU usage and memory shortages  
are signals about systems security

Well-resourced attackers will monitor the system they're attacking to avoid hitting limits or alarms

Experiments can reveal what signals you *should* be collecting – don't take visibility for granted

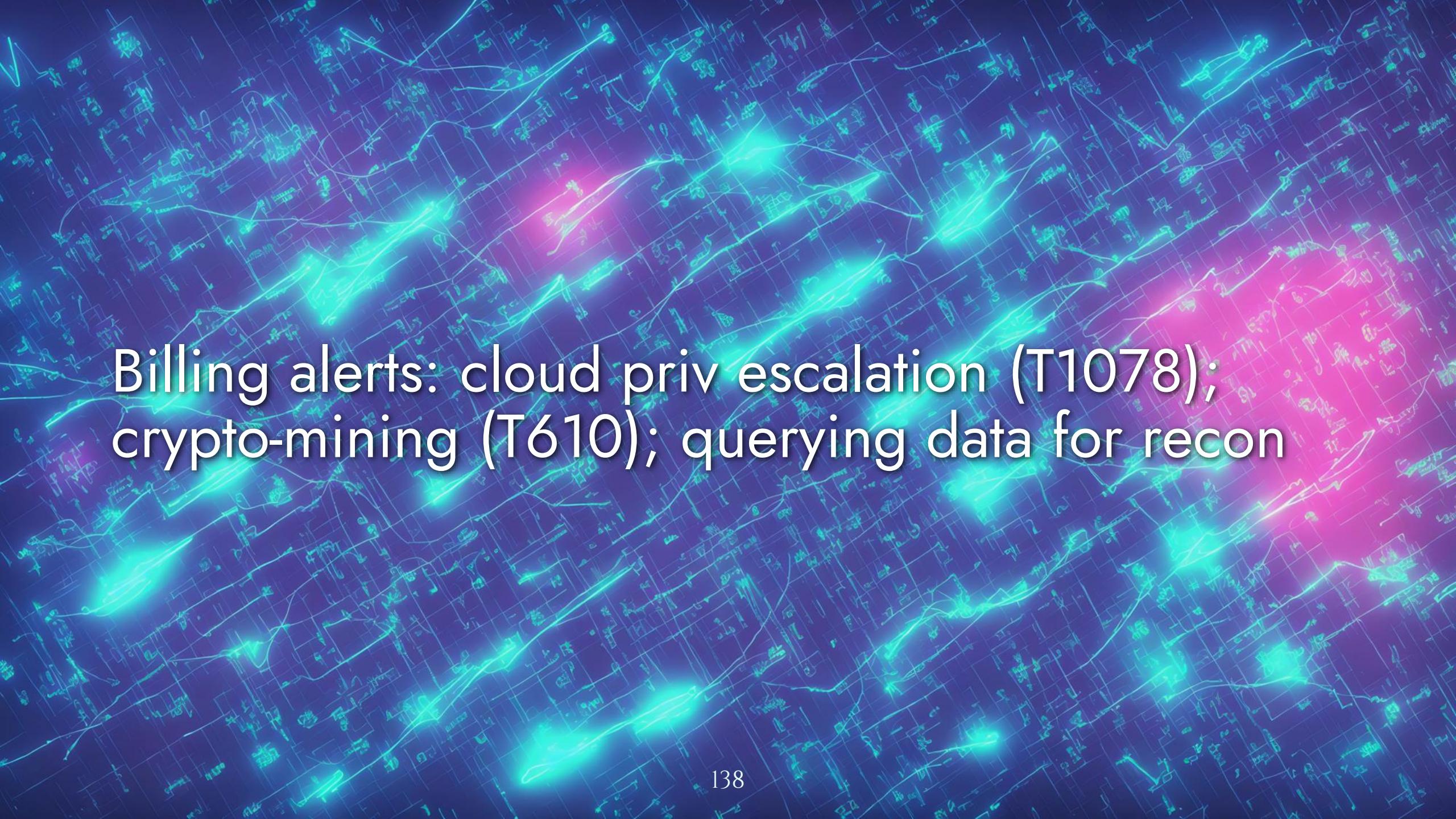
So, what system signals can indicate attacks?  
Turns out SREs and DevOps are our bffs...



Accept queue depth: attacker hijacking system execution (T1574) or process hollowing (T1055)



Autoscale replica count: lateral movement  
(T1072); cryptomining; brute forcing (T1110)

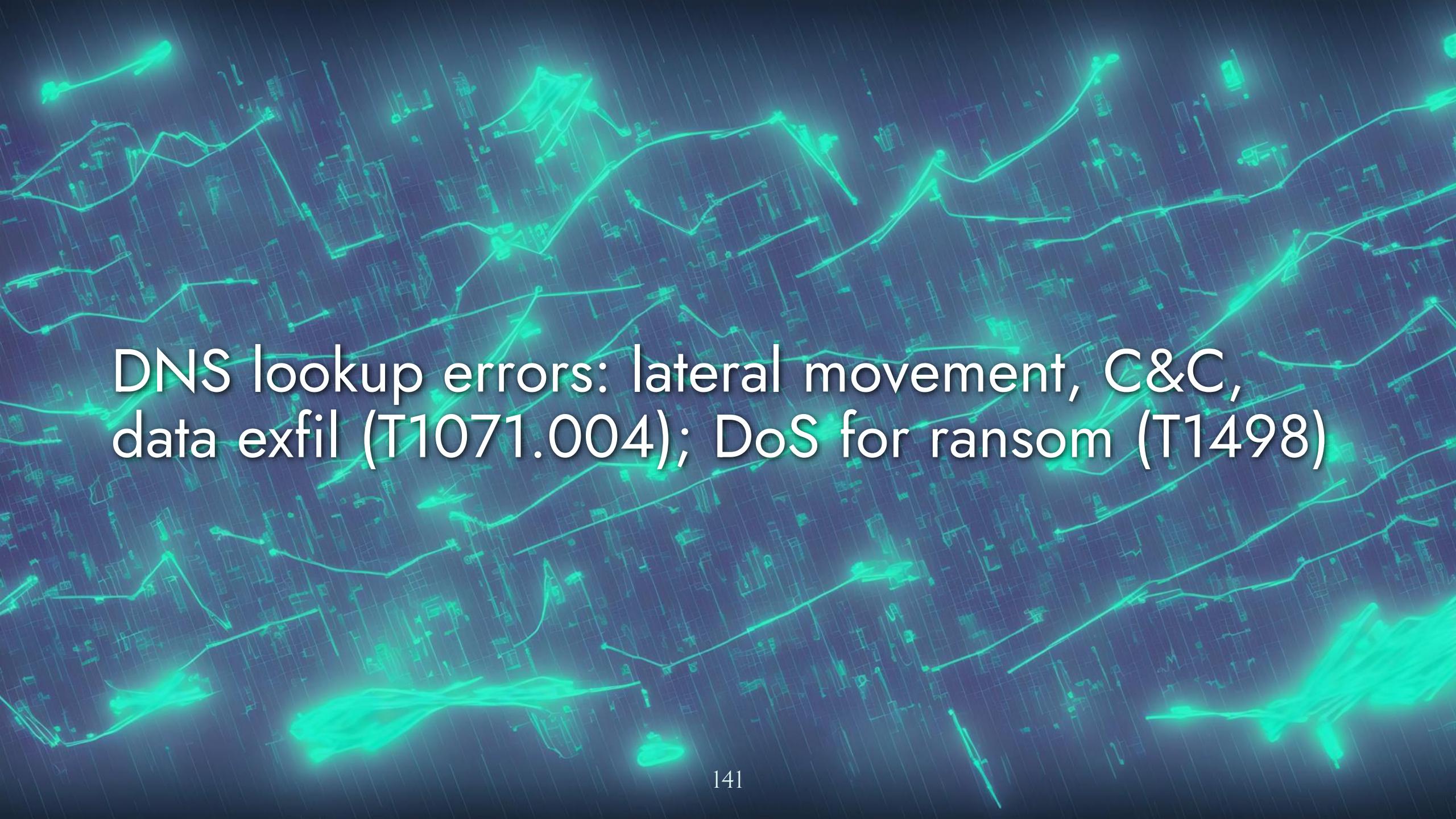


Billing alerts: cloud priv escalation (T1078);  
crypto-mining (T610); querying data for recon

**Cache hit rate (CHR):** DoS; data exfiltration (T1567); brute forcing

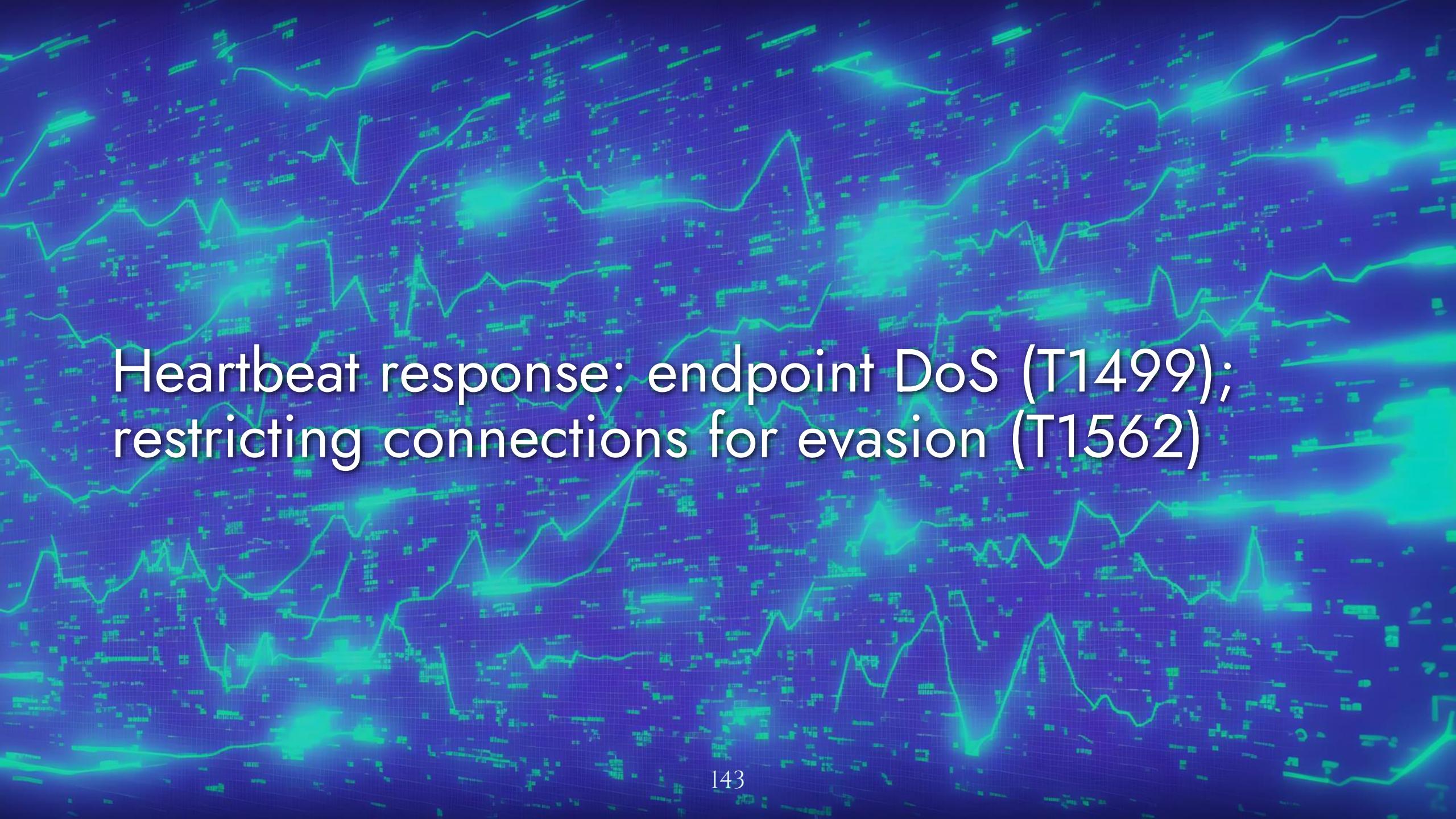


Disk usage, throughput, & IOPS: ransomware  
(T1486); staging data for exfiltration (T1074)



DNS lookup errors: lateral movement, C&C,  
data exfil (T1071.004); DoS for ransom (T1498)

Error rate: credential stuffing (T1110) or DoS

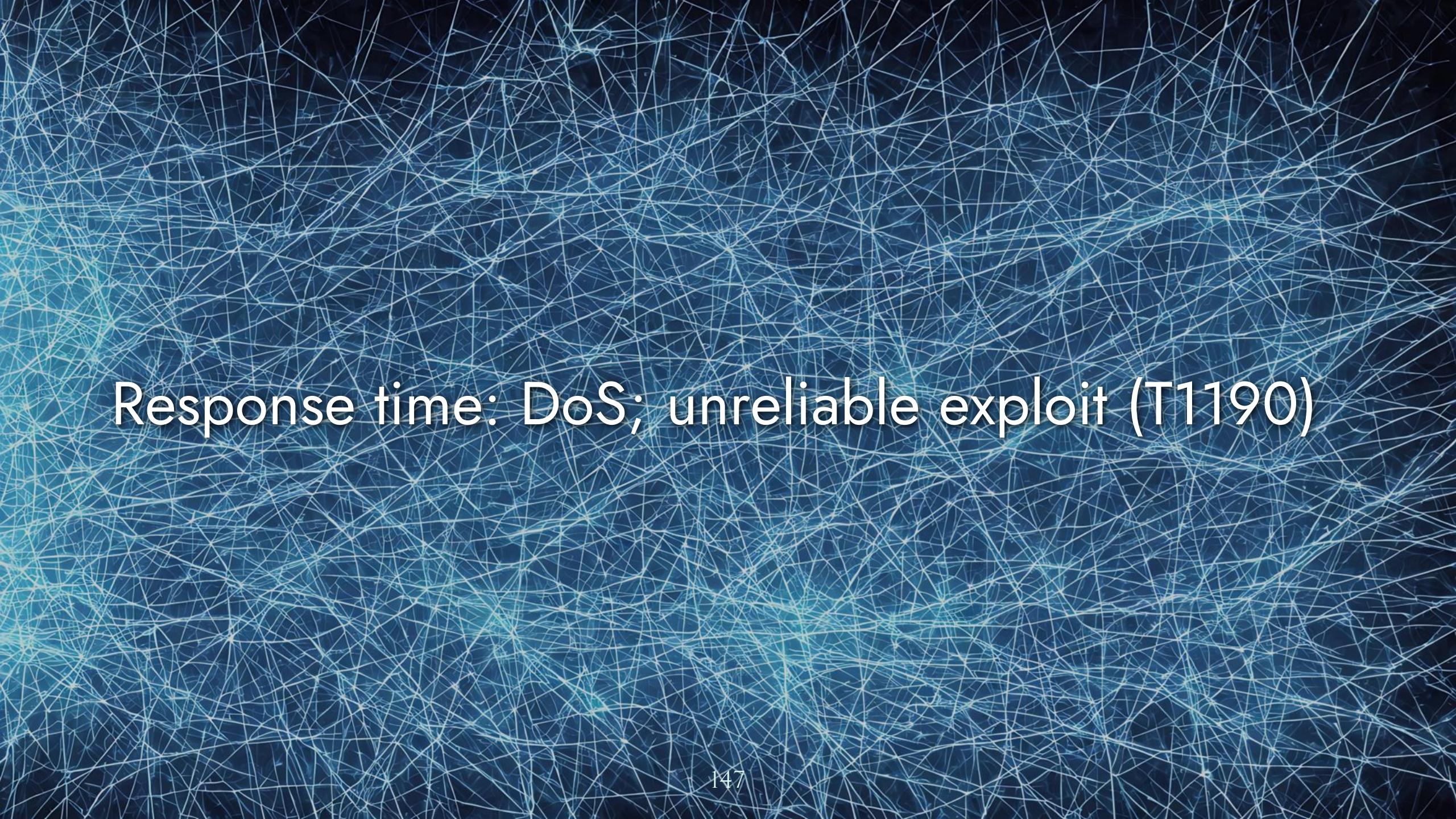


Heartbeat response: endpoint DoS (T1499);  
restricting connections for evasion (T1562)

Rate limit availability: SSRF (T1190); brute force logins (T1110)

Replication lag: unauthorized access or modification (T1565); exploiting inconsistencies

Resource consumption creeping towards max levels (CPU, memory): cryptominers; hijacking resources (T1496); in-memory attacks (T1055)

The background of the slide features a complex, dense network graph. It consists of numerous small, glowing blue dots representing nodes, connected by a web of thin, translucent blue lines representing edges. The graph is highly interconnected, with no single central node, creating a sense of a distributed system or a complex social network.

Response time: DoS; unreliable exploit (T1190)



Swap usage: data exfiltration (T1074.001)

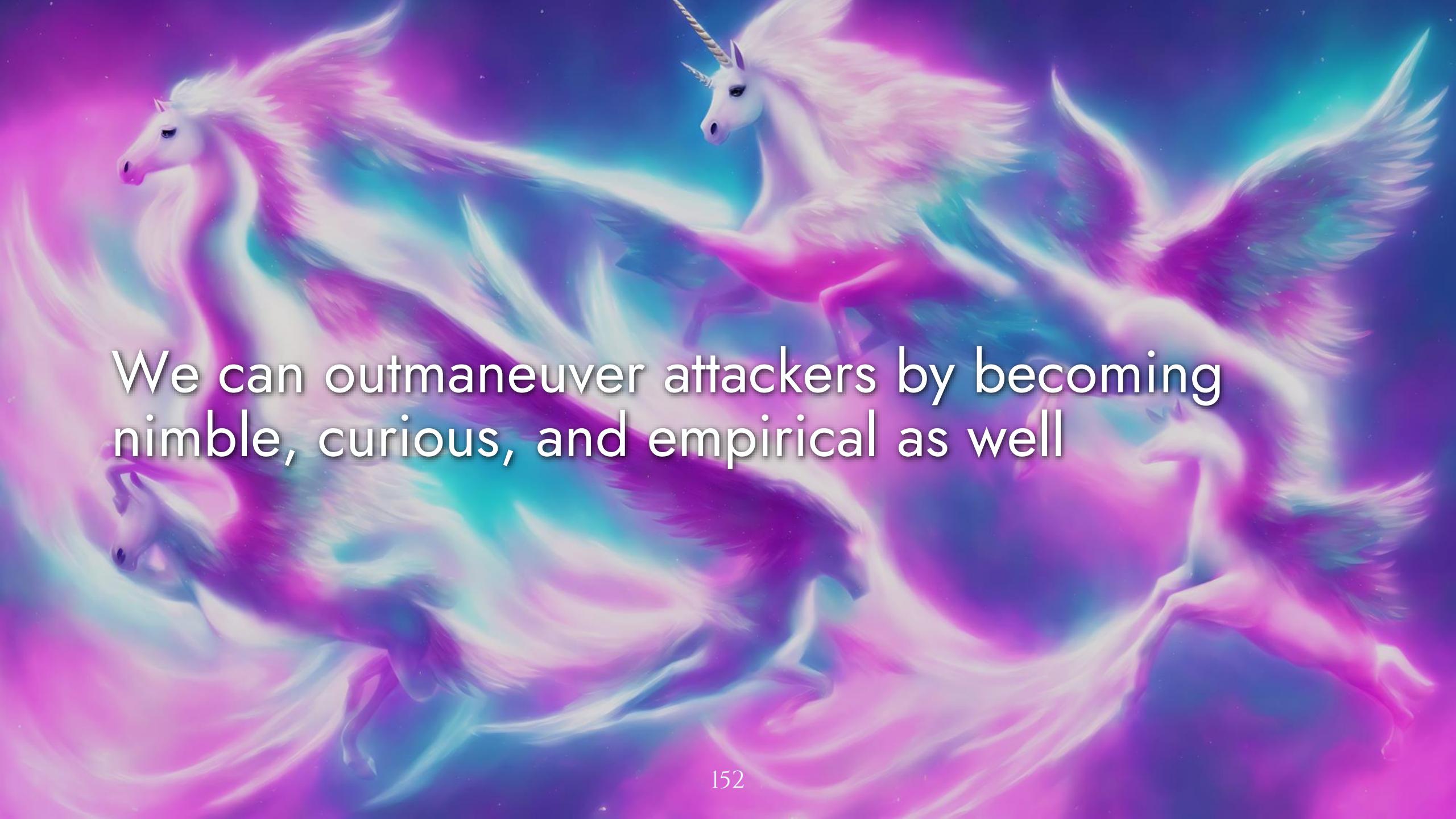


System log lag: stopping or deleting logs to conceal attack operations (T1070)

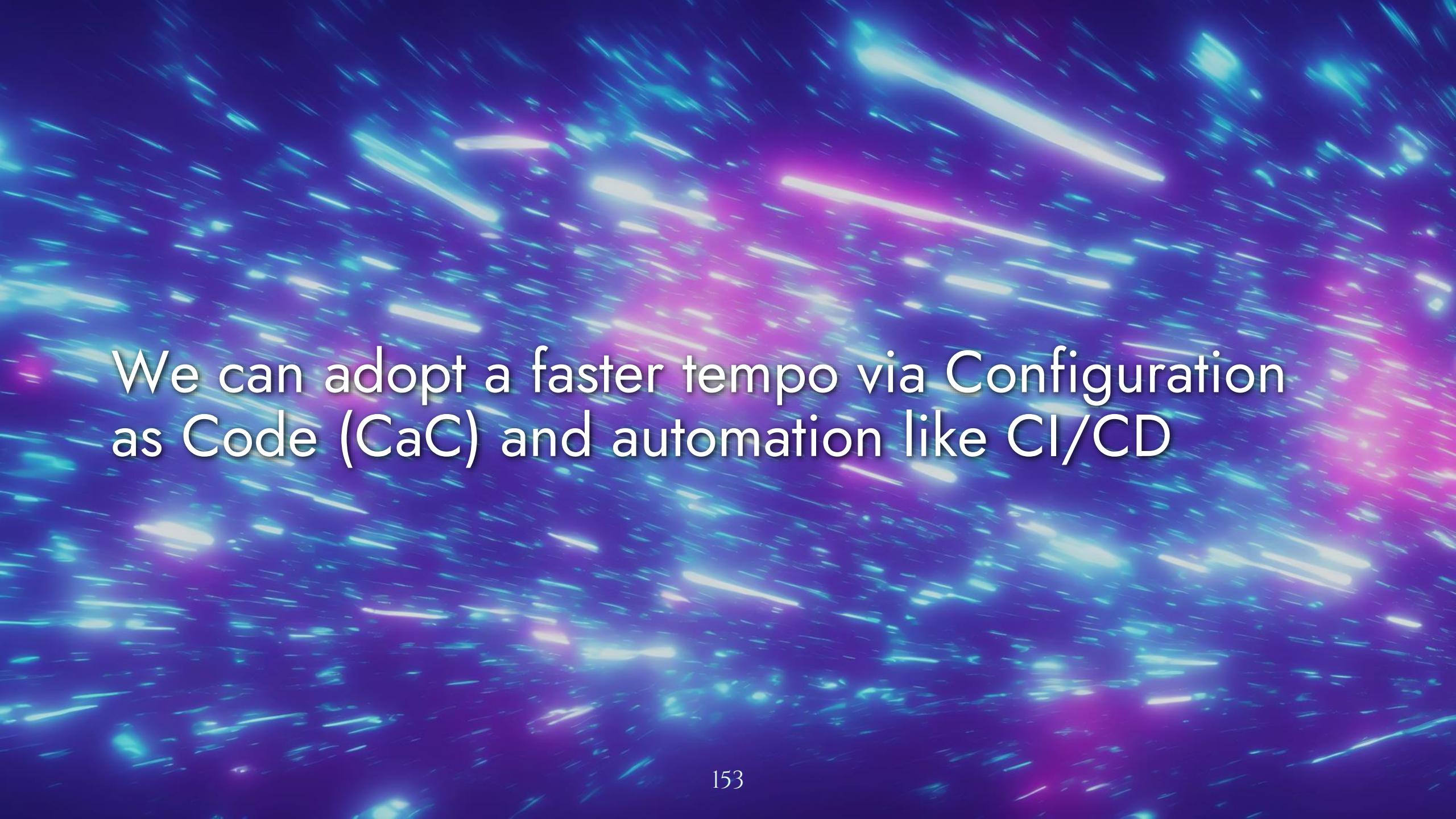
We need our feedback loops to give us immediate sensory input like attackers get



VIVA LAS VEGAS  
LA RÉVOLUTION

A vibrant, abstract background featuring several unicorns and a Pegasus in flight against a colorful, swirling sky.

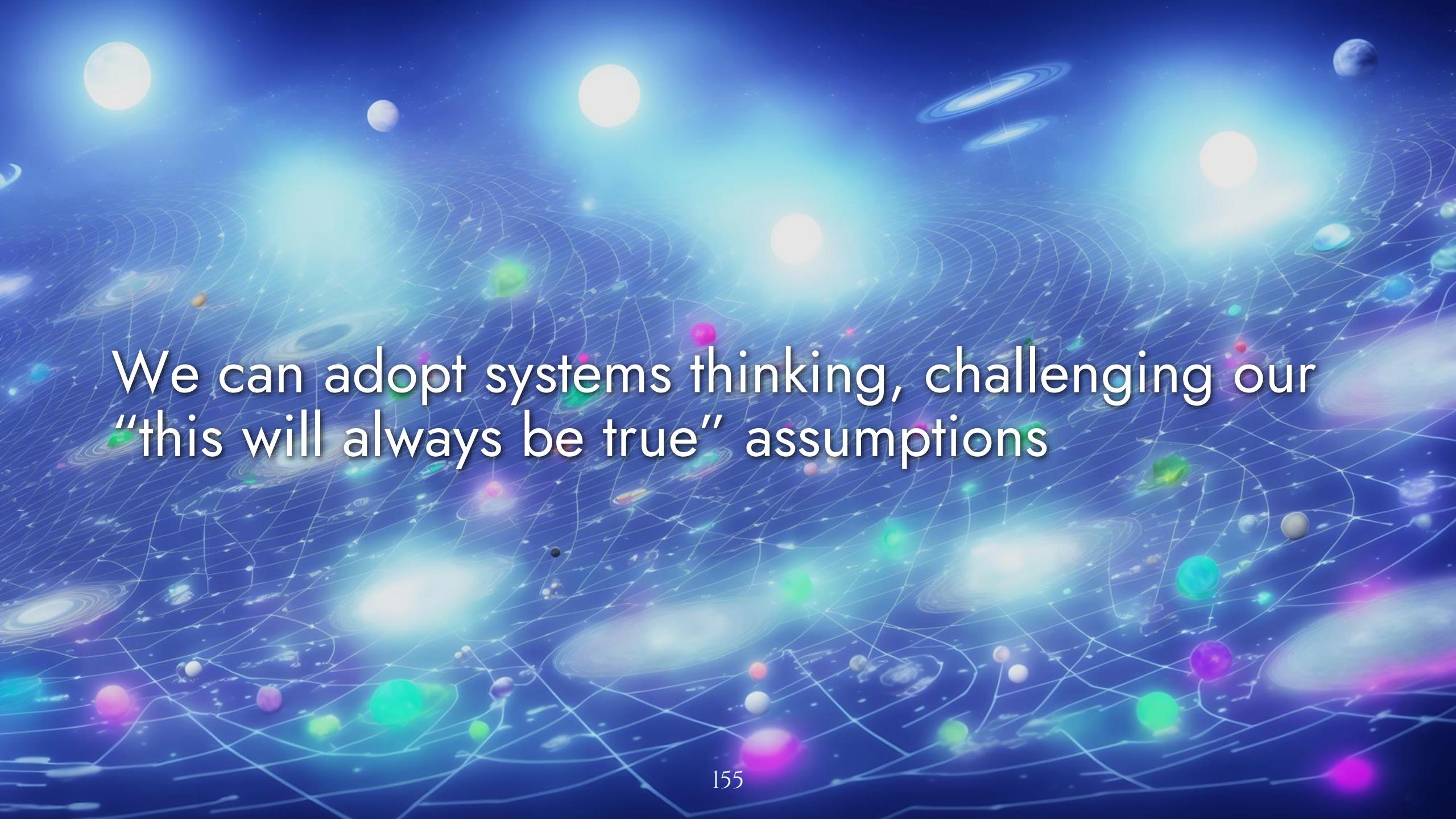
We can outmaneuver attackers by becoming nimble, curious, and empirical as well

The background of the slide features a dynamic, abstract pattern of glowing blue and purple streaks, resembling light trails or data flow, set against a dark background.

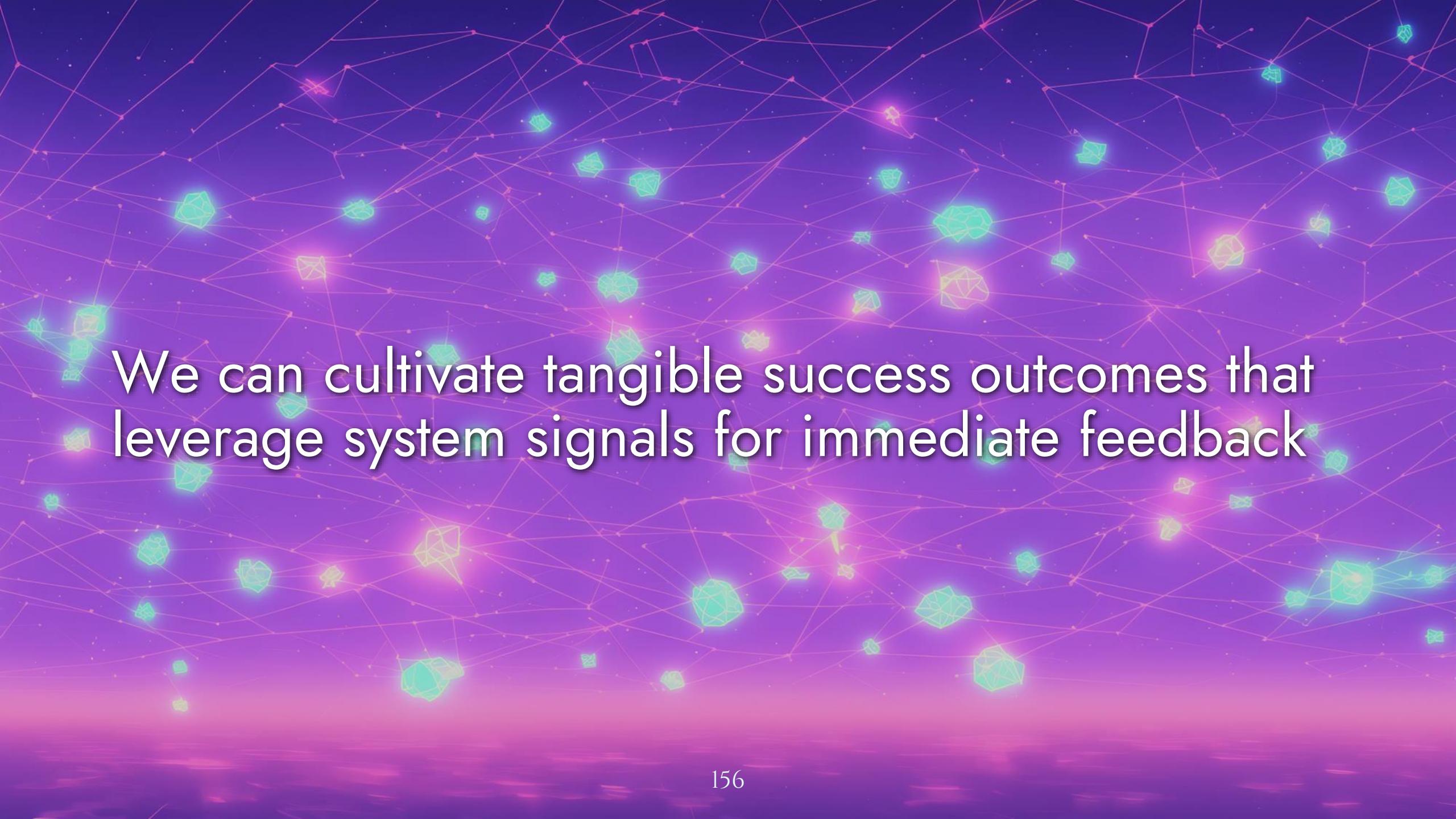
We can adopt a faster tempo via Configuration as Code (CaC) and automation like CI/CD

A futuristic cityscape at night, featuring a winding elevated highway with glowing blue and green lights. Below, a road is paved with glowing pink and teal stones. The city is filled with tall, dark buildings and streetlights, creating a dense urban environment.

We can pursue design-based solutions with our  
Ice Cream Cone Hierarchy and Paved Roads



We can adopt systems thinking, challenging our  
“this will always be true” assumptions



We can cultivate tangible success outcomes that leverage system signals for immediate feedback



We can fuel a feedback loop to gracefully respond to attacks and adapt as attackers evolve



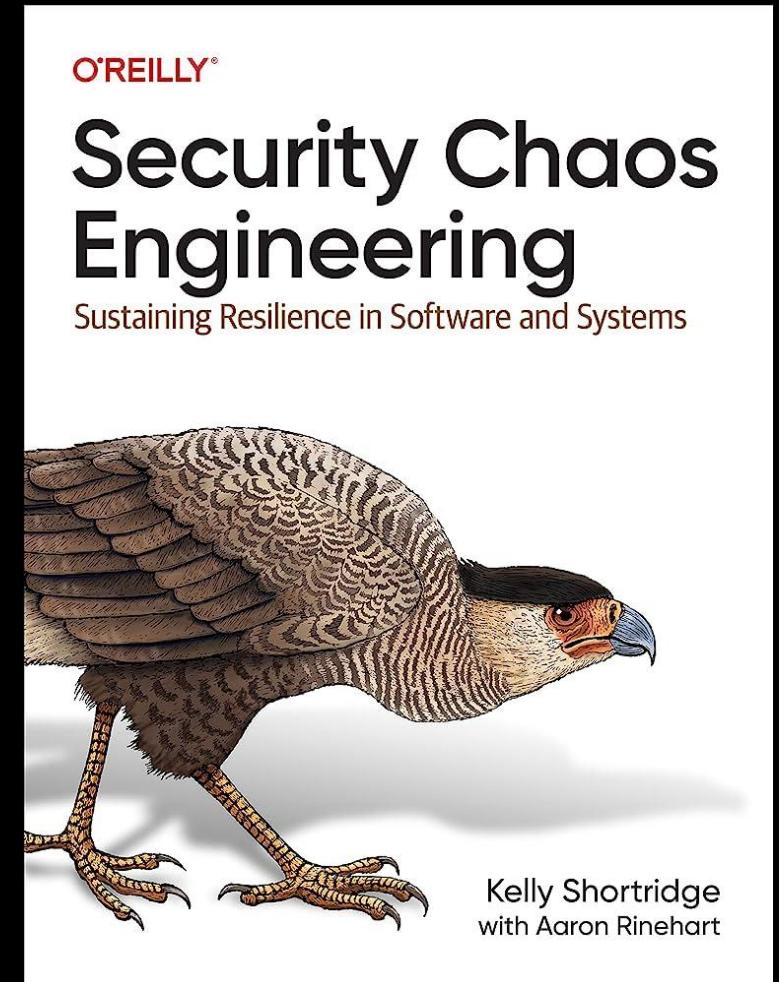
And that, comrades, is the resilience revolution.

Order the book today:

Amazon

Bookshop

& other major retailers





/in/kellyshortridge



@swagitda\_



shortridge@hachyderm.io



@shortridge.bsky.social



chat@shortridge.io