

A wide-angle photograph of a vibrant rainbow arching across a dark, stormy sky. Below the rainbow, the ocean is filled with white-capped waves and spray. In the distance, a small, dark rock formation is visible through the mist.

LAMBOOZLING ATTACKERS

Kelly Shortridge (@swagitda_)

Ryan Petrich (@rpetrich)

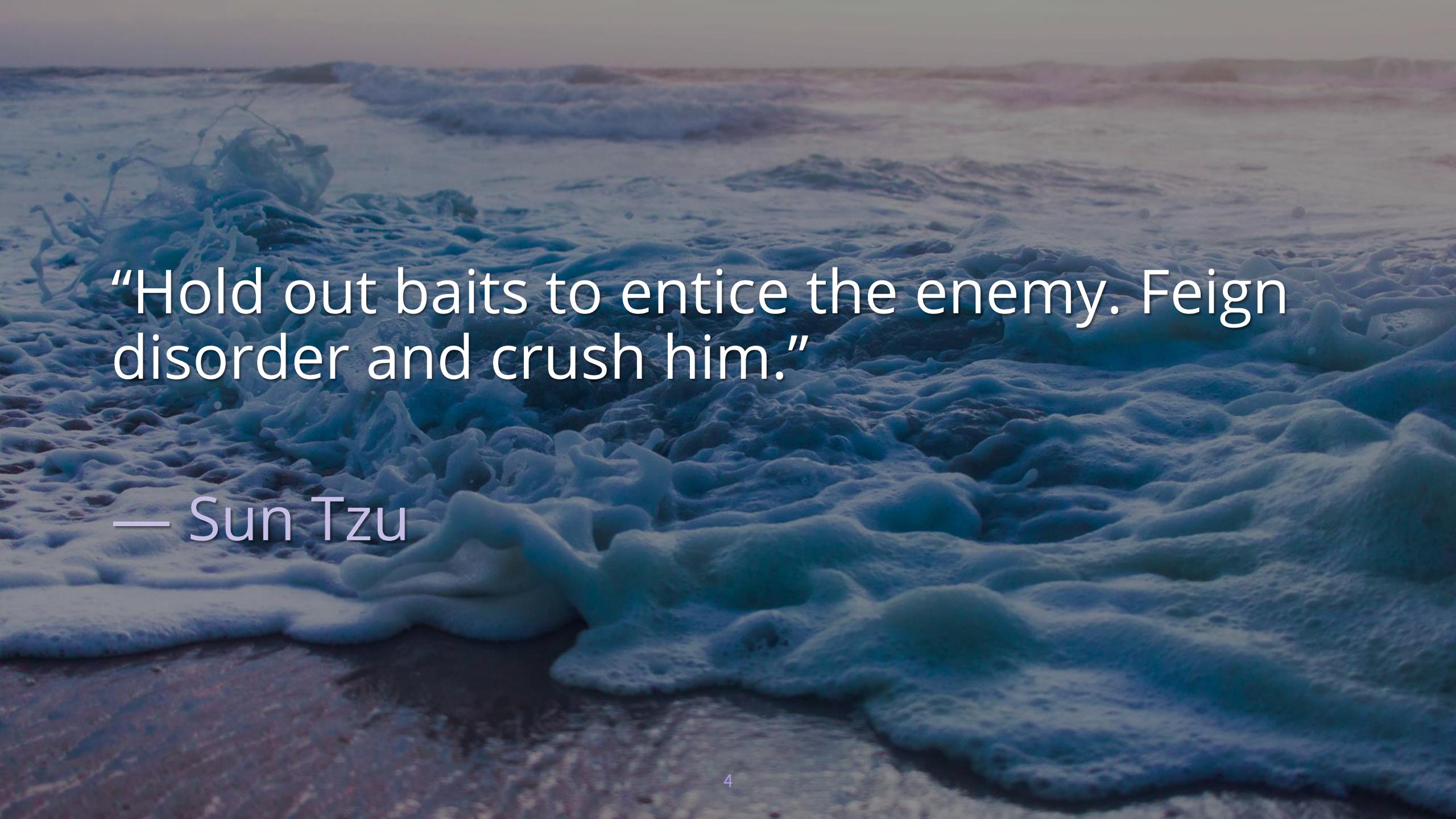
Summercon 2022

A woman with blonde hair, wearing sunglasses and a black bikini, stands in the ocean waves. She is holding a book titled "Security Chaos Engineering" by Aaron Wainhart & Kelly Shortz. The book cover features the O'Reilly logo and the subtitle "Gaining Confidence in Reliability and Safety at Speed and Scale".

Hi, I'm Kelly
fastly

A photograph of a man with dark hair and glasses, sitting on a large, mossy rock at a beach. He is shirtless and wearing dark swim trunks. He is leaning forward with his right hand resting against his chin. Waves are crashing over the rocks behind him, creating white foam. The water is a deep teal color. The sand in the foreground is light brown.

Hi, I'm Ryan

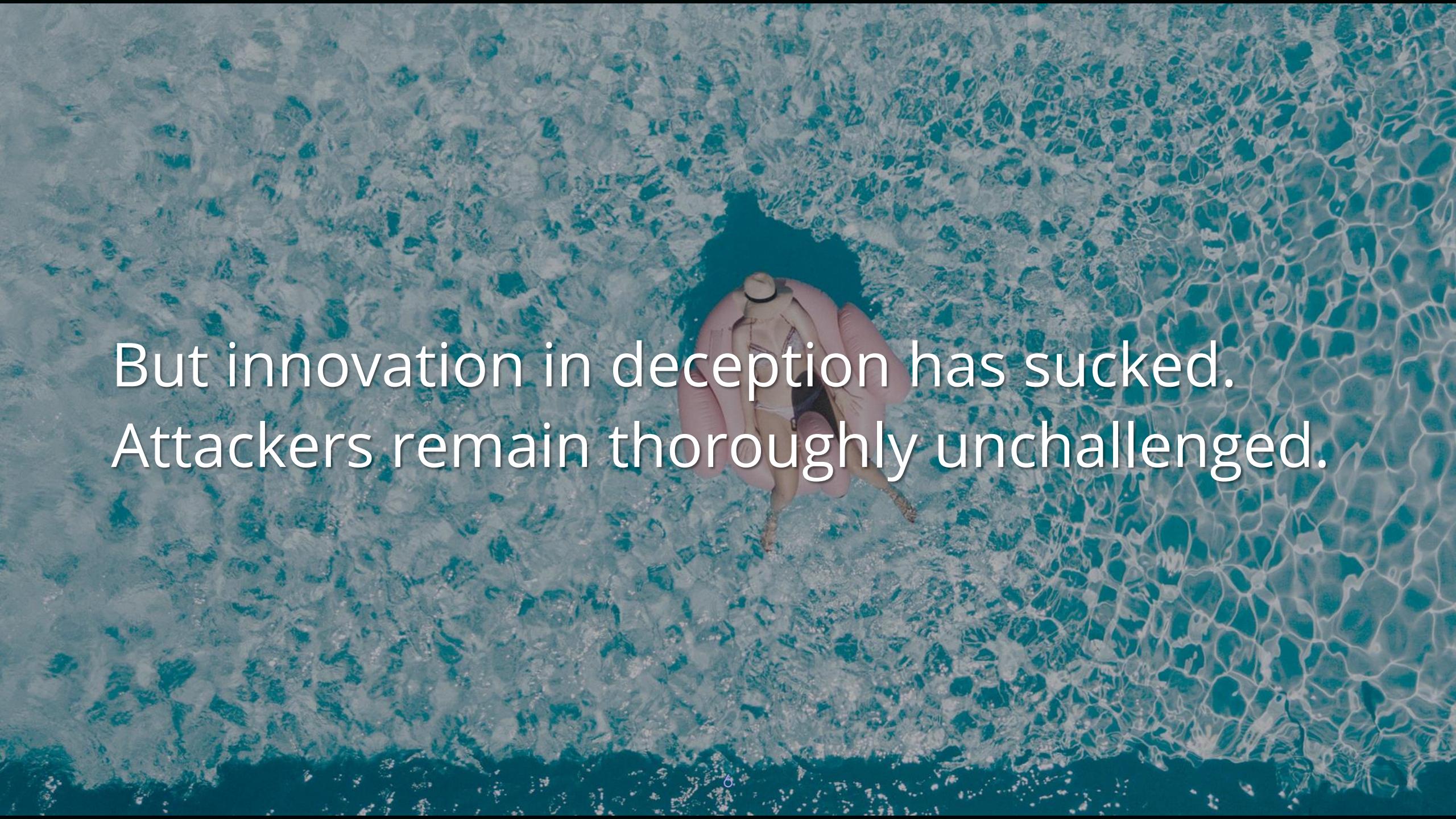


“Hold out baits to entice the enemy. Feign disorder and crush him.”

— Sun Tzu



Deception is a powerful resilience tactic

A photograph of a person in a pink swimsuit floating in clear blue water. The person is positioned centrally, facing downwards, with their arms resting on their head. The water is calm with some ripples, and the background is a bright blue.

But innovation in deception has sucked.
Attackers remain thoroughly unchallenged.



How do we build better deception systems
given our goals, constraints, and tradeoffs?

A photograph of a double rainbow arching across a dark blue sky. Below the sky is a dark teal ocean. The rainbow is bright and clearly visible against the darker background.

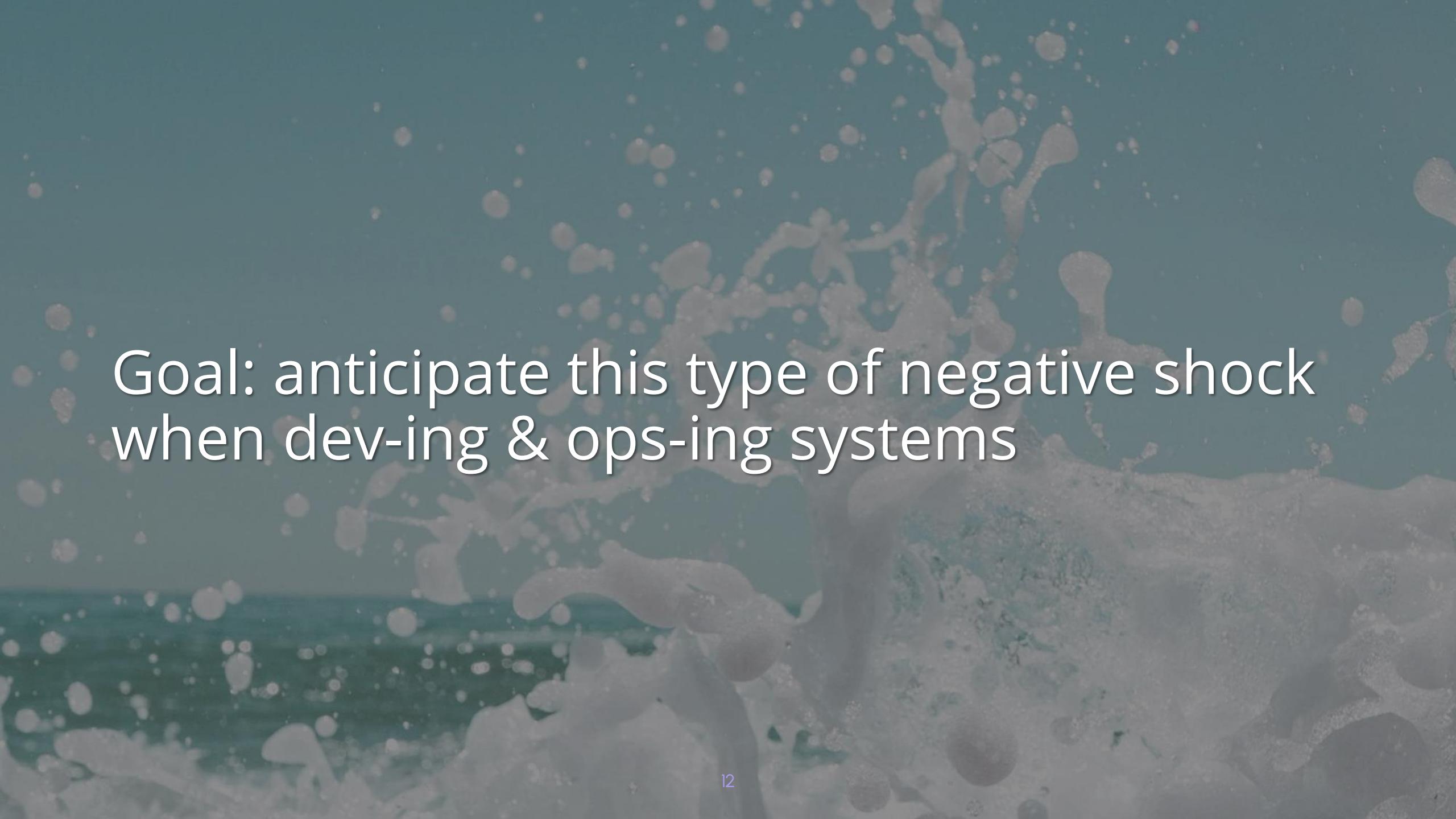
The answer is a new generation of
deception systems: deception *environments*

- I. Exploiting attacker brains
- II. The sucky status quo
- III. Deux ex modern computing
- IV. Designing deception environments
- V. Harvesting potential
- VI. Future opportunities



I. Exploiting attacker brains (for fun & profit)

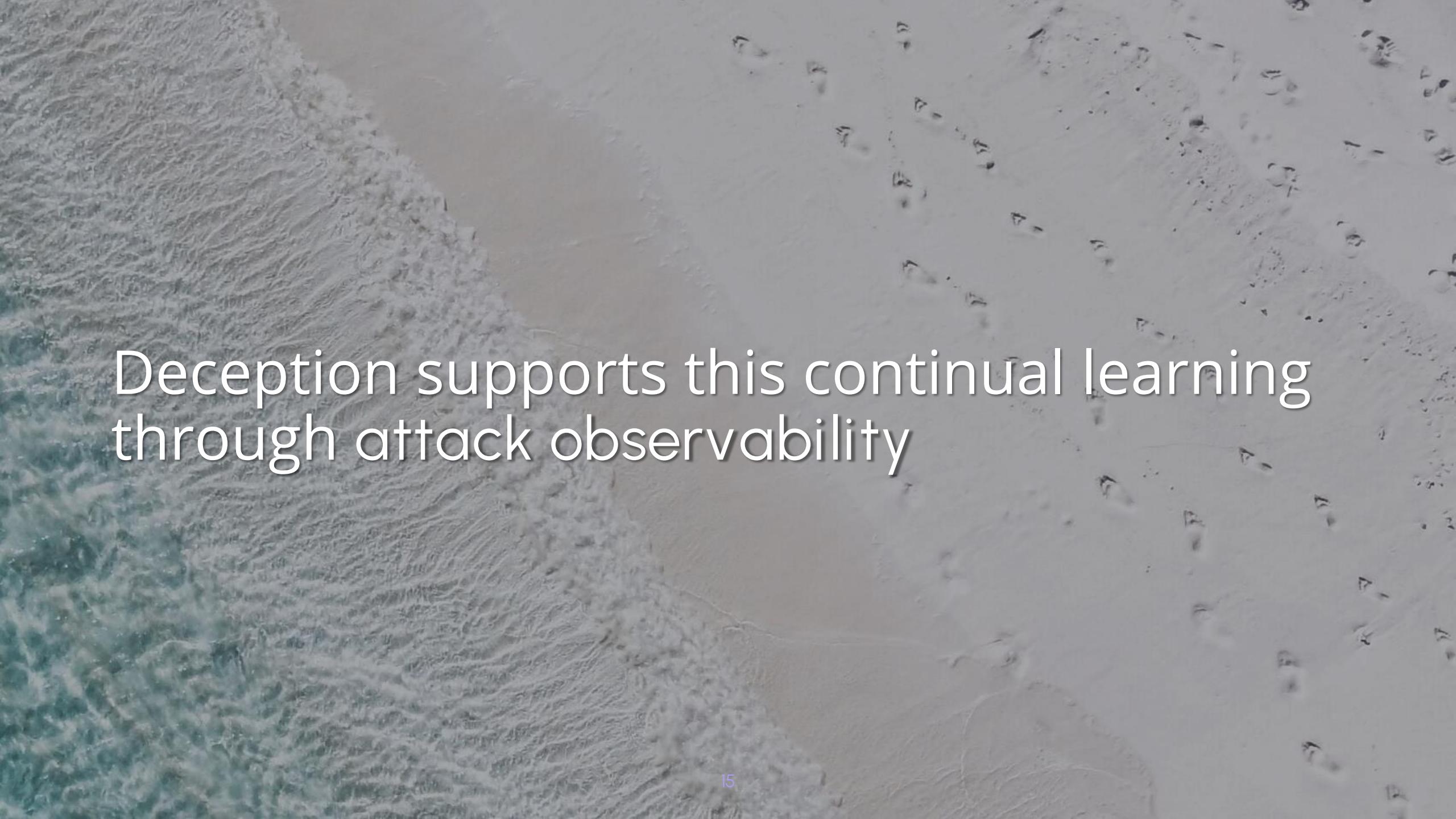
Attackers (plural noun):
humans whose objectives are met by
accessing, destabilizing, stealing, or
otherwise leveraging other humans'
computers without consent



Goal: anticipate this type of negative shock
when dev-ing & ops-ing systems

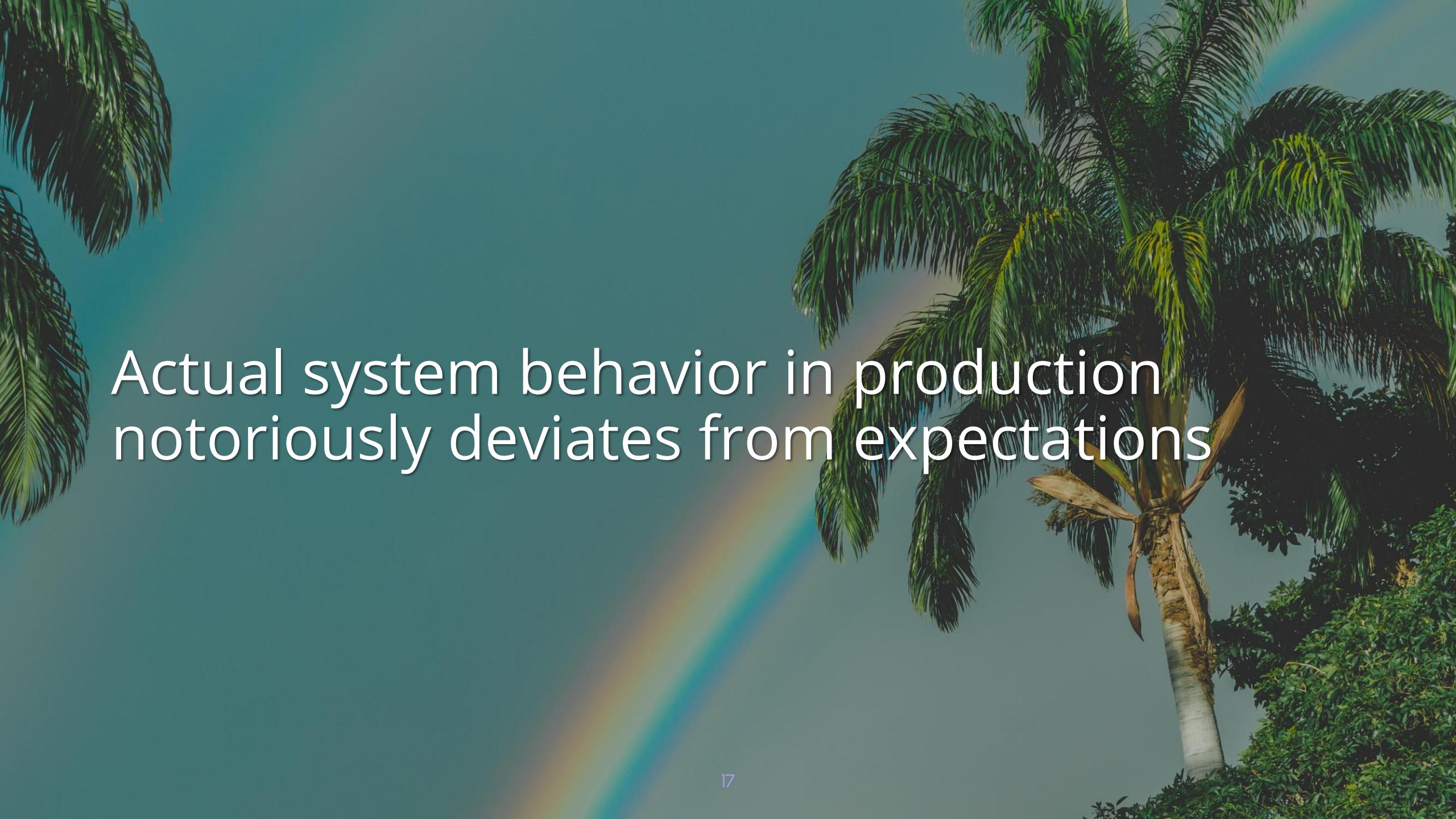
1. Collect relevant info about attackers
2. Implement anticipatory mechanisms that impede the success of attack ops

Sustaining resilience in complex systems
requires a continual learning capacity

The background of the slide is a grayscale aerial photograph of a coastal landscape. It shows a sandy beach in the foreground where waves are breaking, transitioning into darker, more textured ocean water. The horizon line is visible in the distance.

Deception supports this continual learning
through attack observability

Attack Observability: observing the interaction between attackers & systems

The background of the slide features a tropical landscape. On the right side, several palm trees with dense green fronds are visible against a clear blue sky. A bright, multi-colored rainbow arches from the bottom left towards the center of the frame. The overall atmosphere is bright and sunny.

Actual system behavior in production
notoriously deviates from expectations

You may have beliefs about attacker behavior, but does it match reality?

A photograph of a tropical beach. In the foreground, there's a sandy shore with a few palm trees. One prominent palm tree stands on the right side. A bird is captured in flight in the lower-left area. The ocean is a vibrant turquoise color, with white-capped waves crashing onto the shore. The sky is clear and blue.

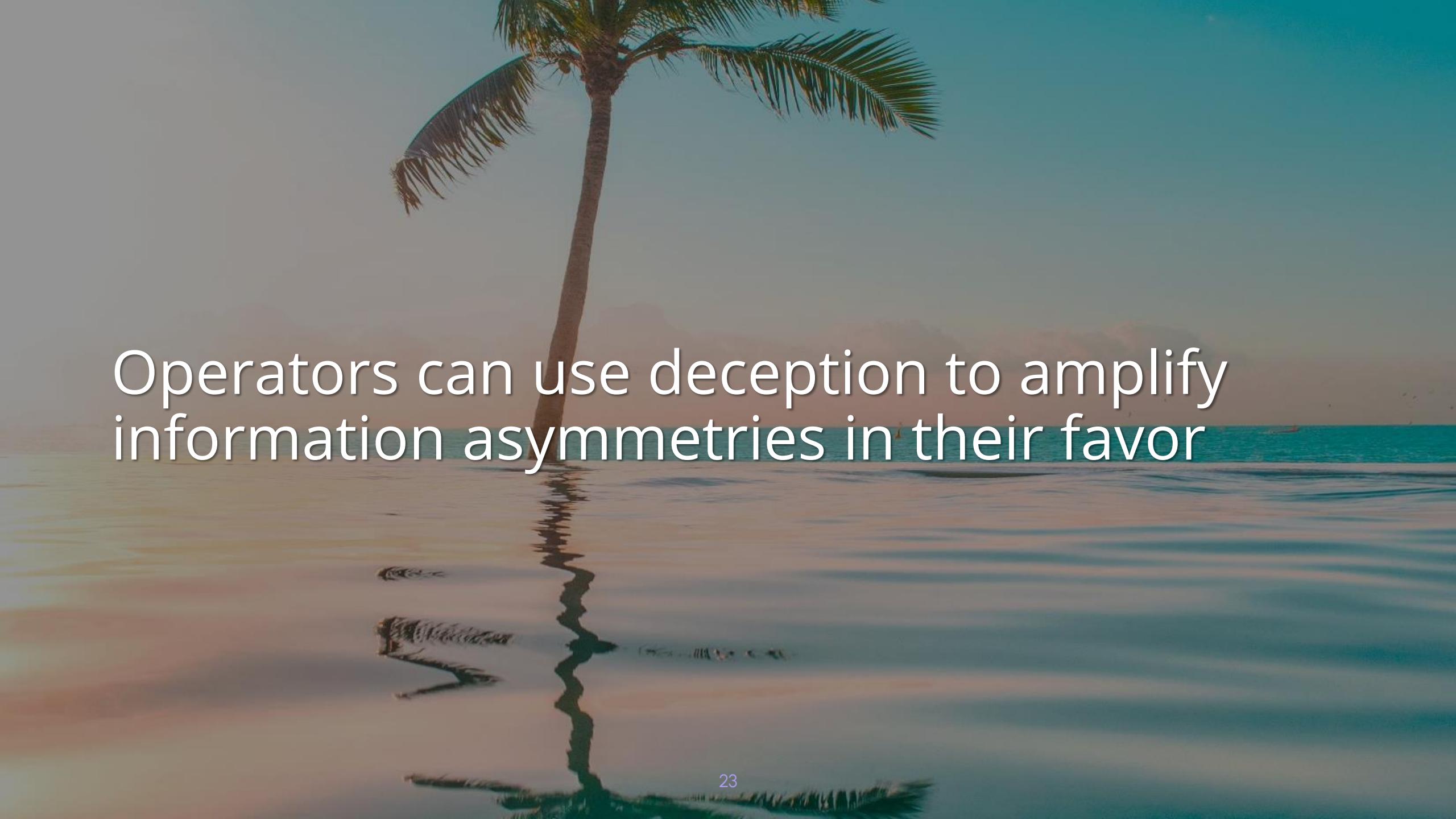
To understand attackers, we need to
understand how humans learn & decide

Human learning & decision-making are
tightly coupled == exploit opportunity

A photograph of two palm trees silhouetted against a blue sky. The sky has a gradient from light blue to a warm orange and pink near the horizon, suggesting a sunset or sunrise. The palm trees are positioned in the lower half of the frame, with one on the left and one on the right.

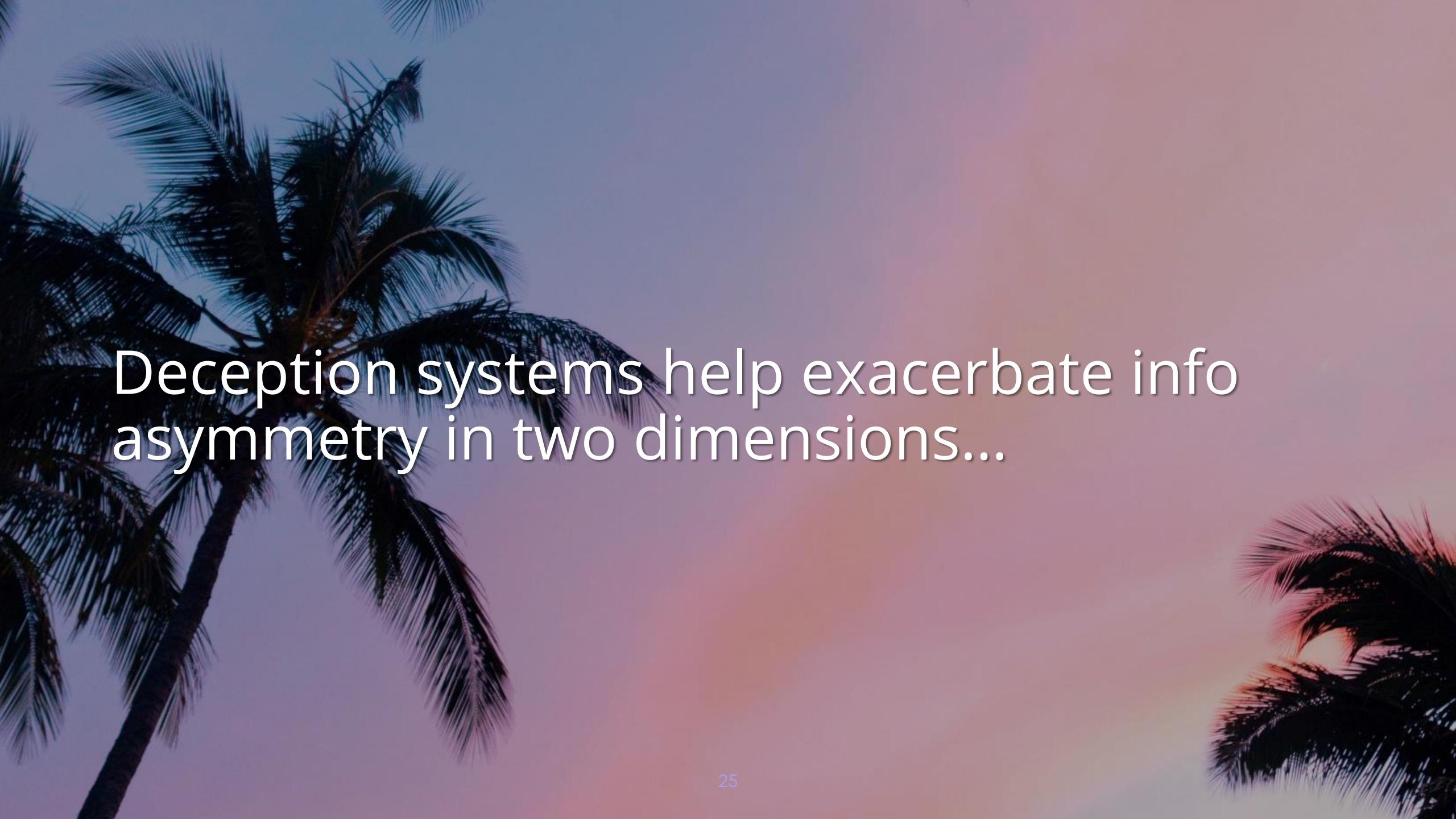
Information asymmetry leads to core
advantages for one “side” of the game

Each side chooses a plan based on pre-existing beliefs + learned experience

A photograph of a tropical beach at sunset. A single palm tree stands prominently on the left, its silhouette and fronds reflected perfectly in the calm, blue-tinted water. The sky is a gradient of warm orange and yellow near the horizon, transitioning to a cooler blue as it meets the sea. In the distance, faint outlines of hills or mountains are visible under the setting sun.

Operators can use deception to amplify
information asymmetries in their favor

Make attacker experiences unreliable;
poison the attacker's learning process

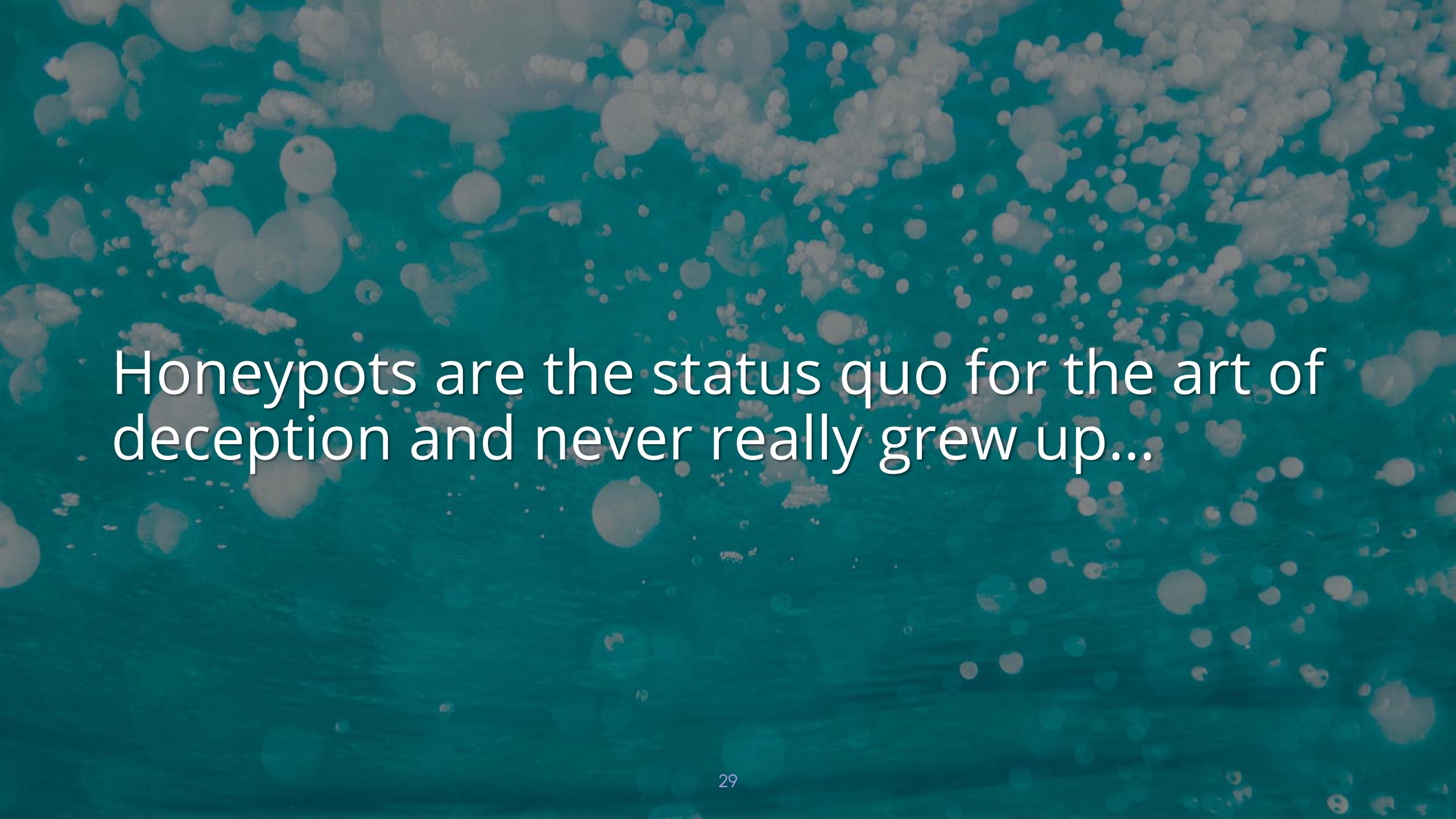
The background of the slide features a photograph of several palm trees silhouetted against a vibrant sunset. The sky transitions from a deep blue at the top to warm shades of orange, red, and yellow near the horizon. The palm fronds are dark and textured, creating a tropical atmosphere.

Deception systems help exacerbate info asymmetry in two dimensions...

1. Expose real-world data on attackers' thought processes (increasing the value of info for operators)

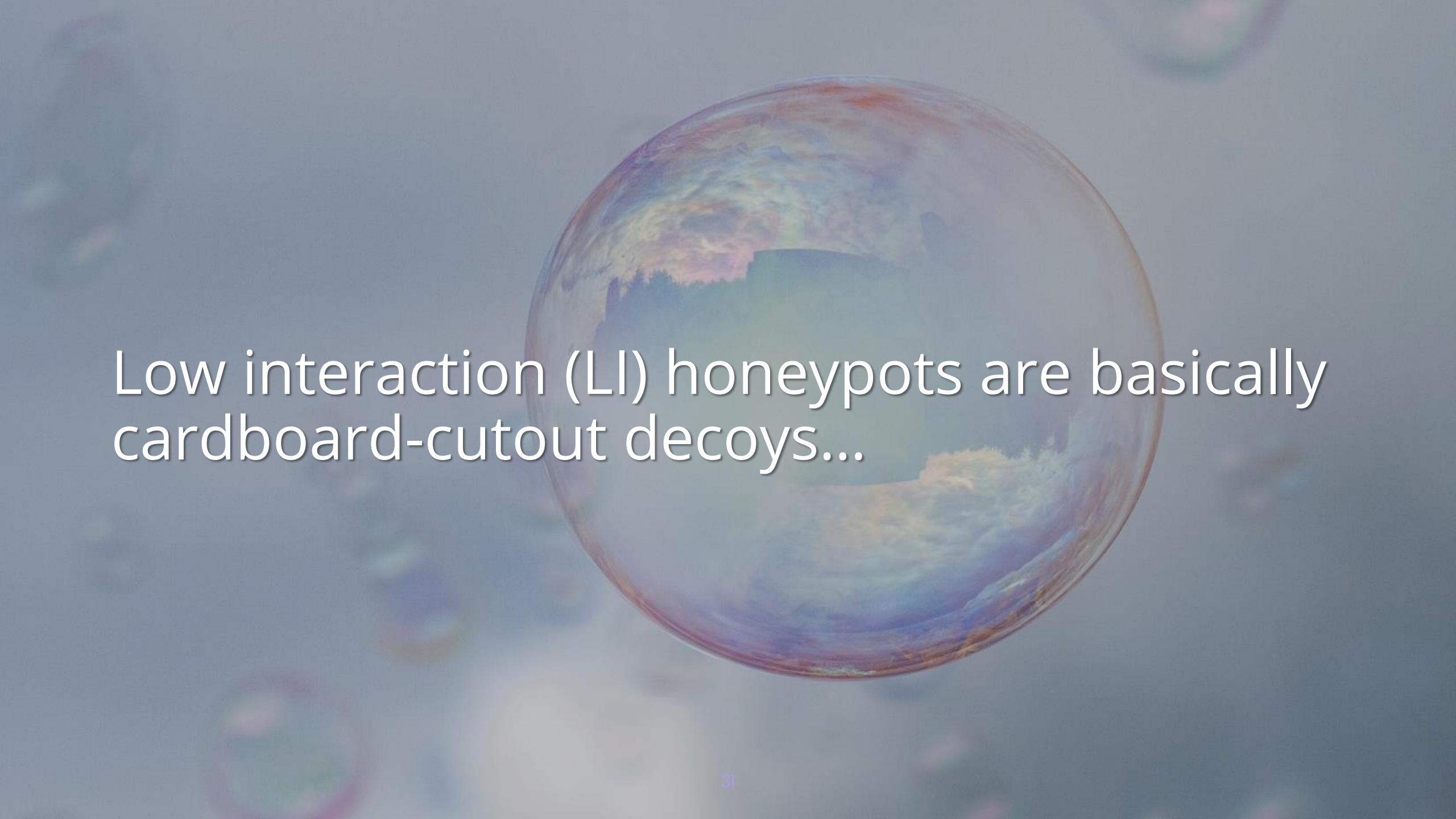
2. Manipulating info to disrupt attackers' abilities to learn & make decisions
(reducing the value of info for attackers)

III. The sucky status quo



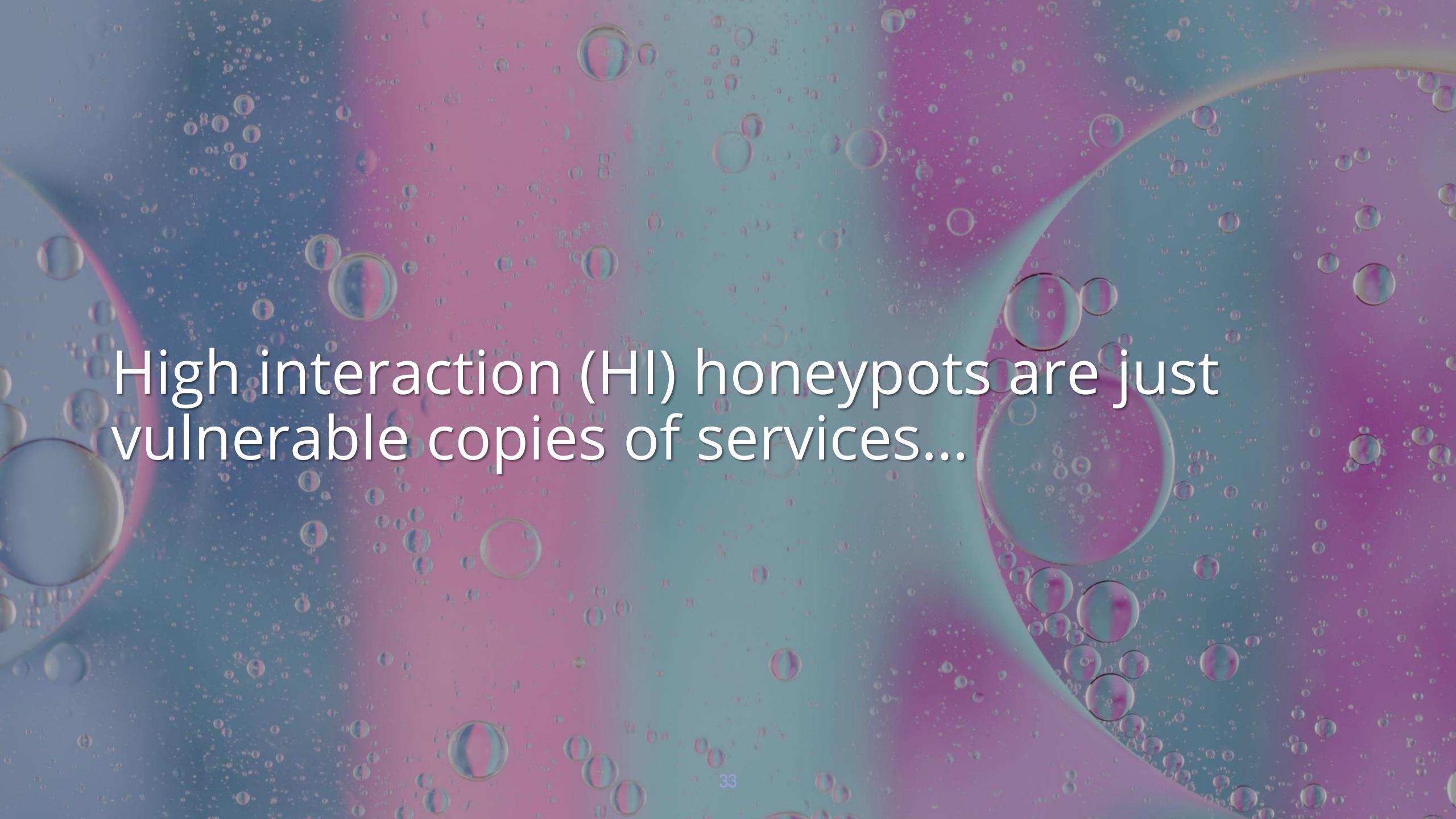
Honeypots are the status quo for the art of deception and never really grew up...

What kinds of honeypots are there?



Low interaction (LI) honeypots are basically
cardboard-cutout decoys...

Medium interaction (MI) honeypots imitate a specific system without meaningful depth

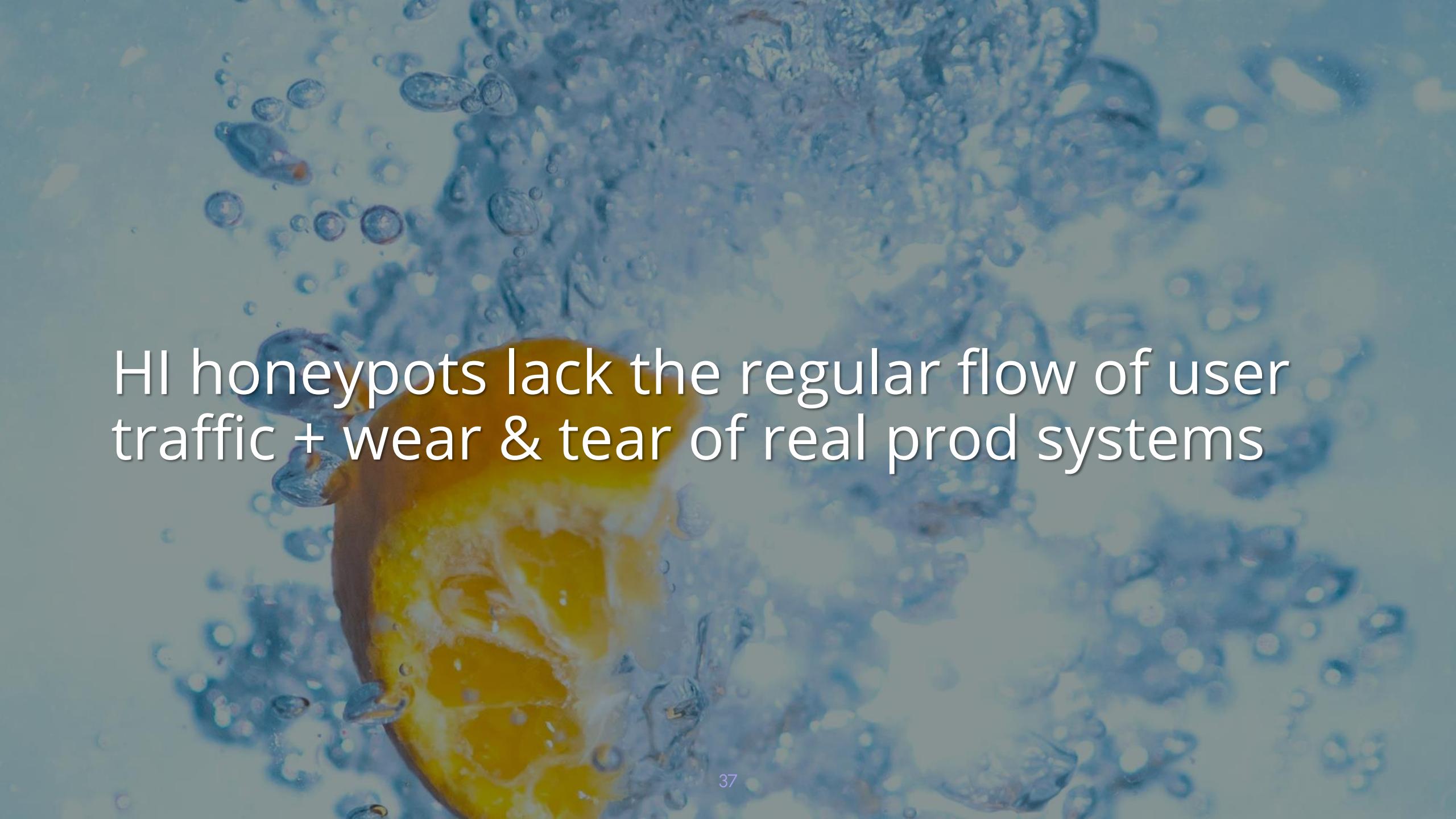
The background of the slide features a dense, abstract pattern of numerous small, semi-transparent oil bubbles. These bubbles are scattered across the frame, with larger ones appearing towards the edges and corners. They reflect a variety of colors, including shades of blue, green, yellow, and pink, creating a vibrant, liquid-like texture.

High interaction (HI) honeypots are just
vulnerable copies of services...

LI & MI honeypots are ineffectual af at
deceiving attackers so we can dismiss them

Even HI honeypots are unconvincing to attackers with a modicum of experience

“Does the system feel real?” (no)
“Does it lack activity?” (yes)



HI honeypots lack the regular flow of user traffic + wear & tear of real prod systems

P.S. a fundamental flaw of honeypots is
that they're controlled by infosec people...



III. Deux ex modern computing

We really *need* a new generation of
deception given its potential for resilience



Deception Environments are this new gen
and differ both in design & ownership

Attackers have expertise in attacking
systems – so no wonder the status quo fails

A photograph of a tropical beach at sunset. The sky is filled with large, billowing clouds colored in shades of orange, yellow, and blue. The ocean waves are crashing onto the light-colored sand. In the distance, several palm trees stand on the shore. The overall atmosphere is peaceful and scenic.

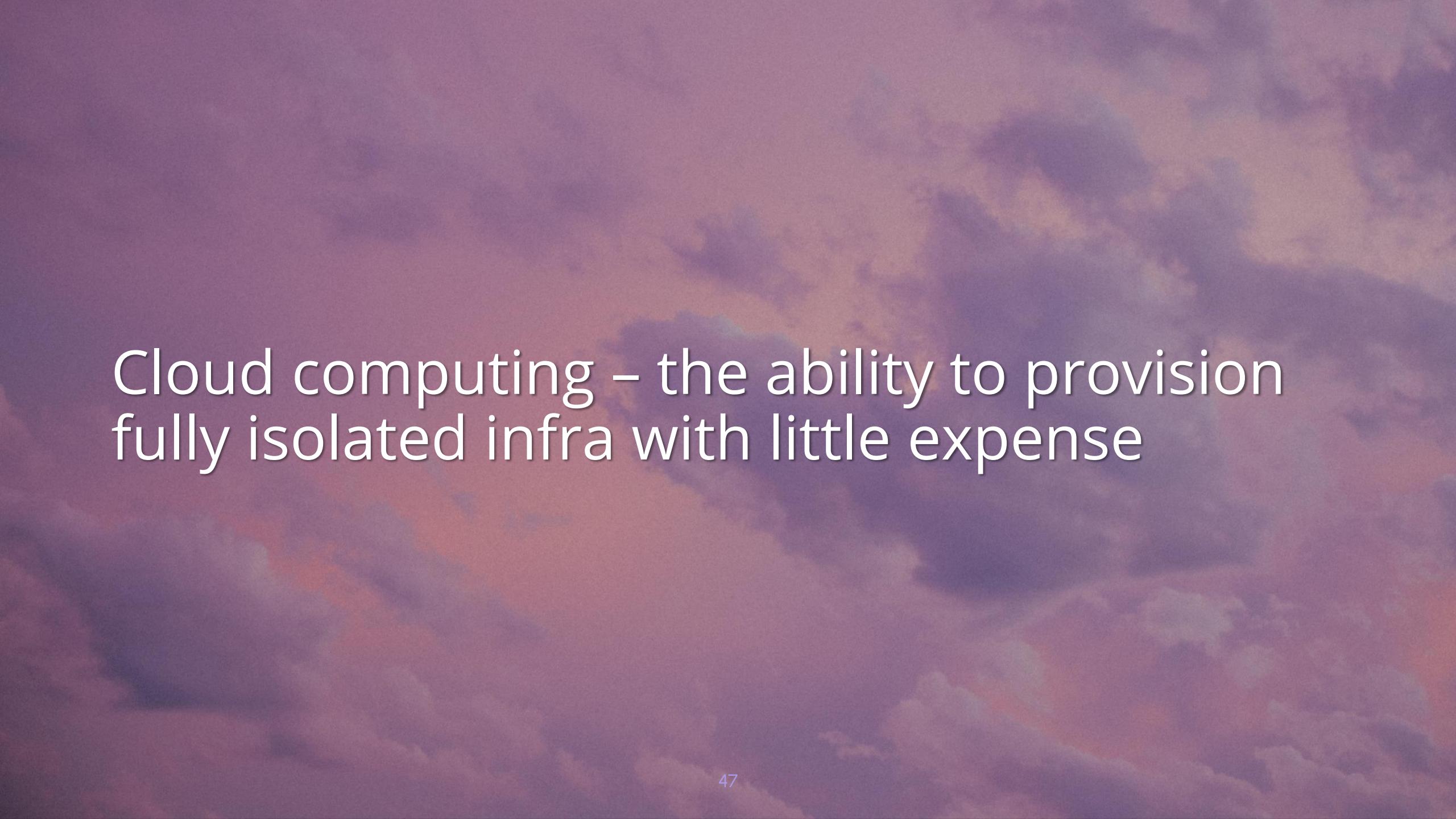
*Deception environments (DEs) are possible
with new types of computing + new owners*

Goal of traditional honeypots = frequency
of scanning tools or exploiting known vulns

DEs observe attacker behavior through all operational stages + experiment on them



What parts of modern infra help lower costs & improve deception design efficacy?

The background of the slide features a dramatic sky at sunset or sunrise. The clouds are thick and textured, with colors ranging from deep purple and blue to bright orange and yellow. The light is low, creating a warm glow and casting long shadows.

Cloud computing – the ability to provision
fully isolated infra with little expense

Deployment automation + defining infra declaratively decreases ops overhead



Virtualization advancements: isolation,
observability, denser computing

SDN proliferation enables isolated network topology dedicated to attackers

The background of the slide features a wide-angle photograph of a sky at dusk or dawn. The upper portion of the sky is a deep, clear blue. Below it, there are several layers of clouds. The most prominent layer consists of large, puffy cumulus clouds that are heavily tinged with warm colors like orange, red, and yellow, suggesting the light of the sun is hitting them from behind. In the lower-left foreground, there are darker, more shadowed clouds, possibly stratus or cumulus clouds, which provide a nice contrast to the brightly lit upper clouds.

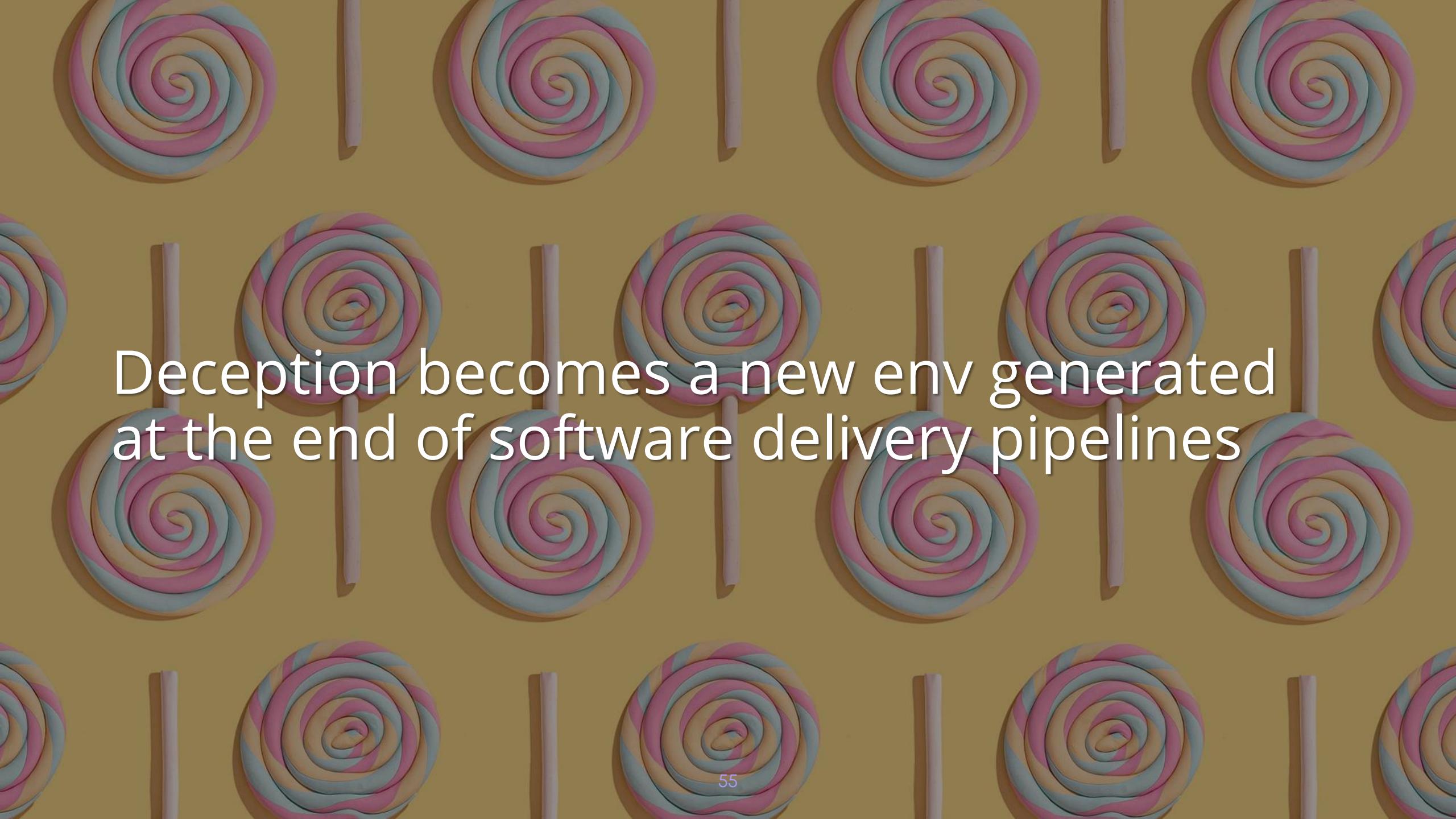
Ownership should be based on systems
design expertise, not security expertise

SWEs can repurpose deployment templates
to build unique, powerful deception envs



IV. Designing deception environments

DE design philosophy: repurpose the
design, assets, & templates of a real system



Deception becomes a new env generated
at the end of software delivery pipelines

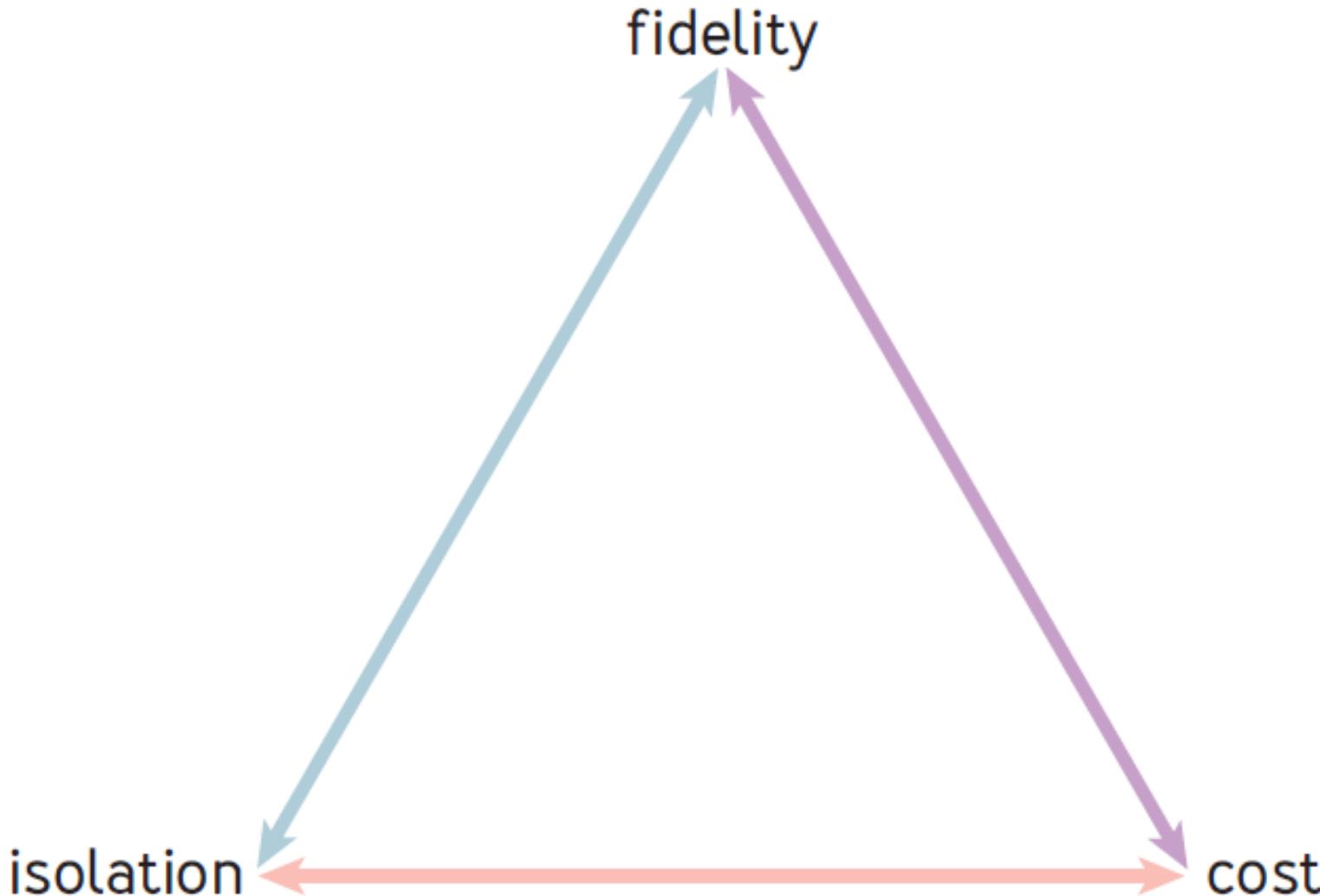
We can preempt attacker skepticism by
designing a DE that feels “lived in”



Starting with the design of a real prod
system == realism + more relevant insights

The F.I.C. trilemma: fidelity, isolation, cost

FIGURE 1: THE FIC TRILEMMA FOR DECEPTION SYSTEMS

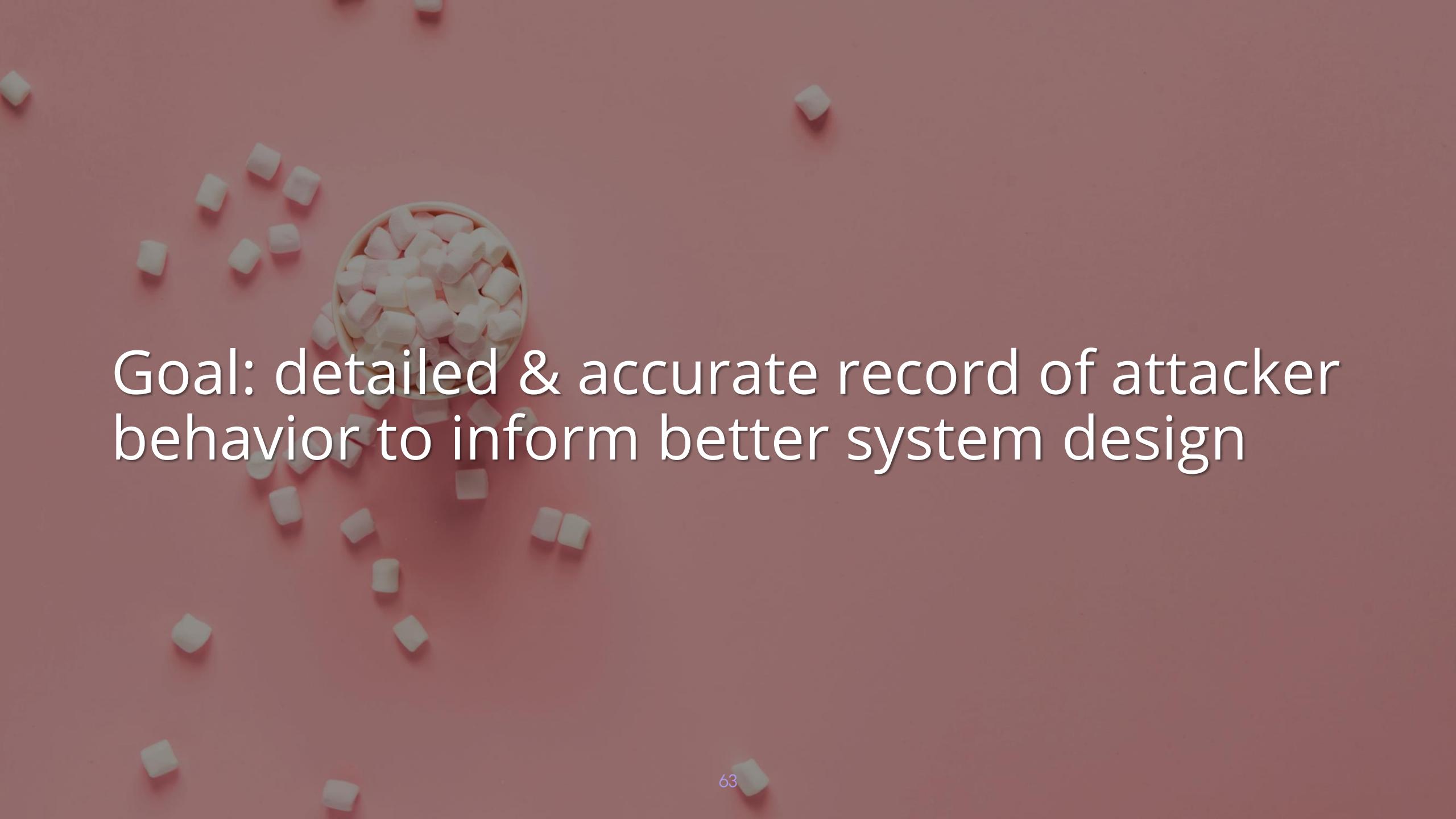


Fidelity: credibility to attackers and ability
to support attack observability

A close-up photograph of a vibrant, multi-layered cake. The cake features several distinct layers of different colors, including yellow, pink, and purple, separated by white frosting. Various decorative elements are visible, such as a small pink fortune cookie and a piece of candy with blue and red markings. The cake is set against a dark background.

Attackers expect to see things like a service running, prod-like traffic, coordinating with other services, orchestration, monitoring...

Deception systems need sufficient depth to stimulate extended attacker activity



Goal: detailed & accurate record of attacker behavior to inform better system design

Isolation: degree to which the deception system is isolated from the real env or data



Goal: not jeopardizing availability of the real system or data privacy

Cost: computing infra + operational overhead required to deploy & maintain

Goal: minimal operational burden;
expensive means more unlikely to be used



FIGURE 2: EXAMPLE DECEPTION SYSTEMS MAPPED TO THE FIC TRILEMMA

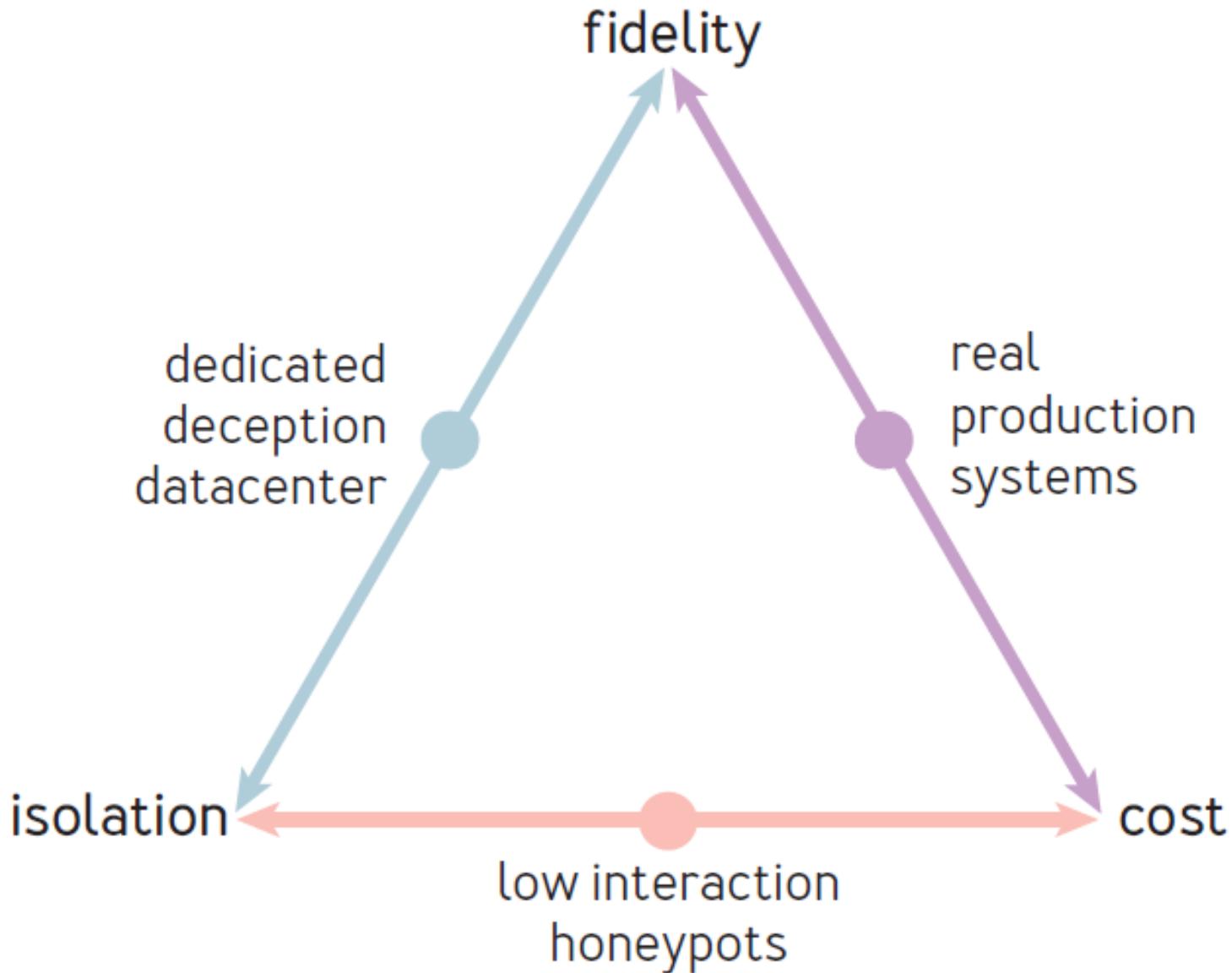
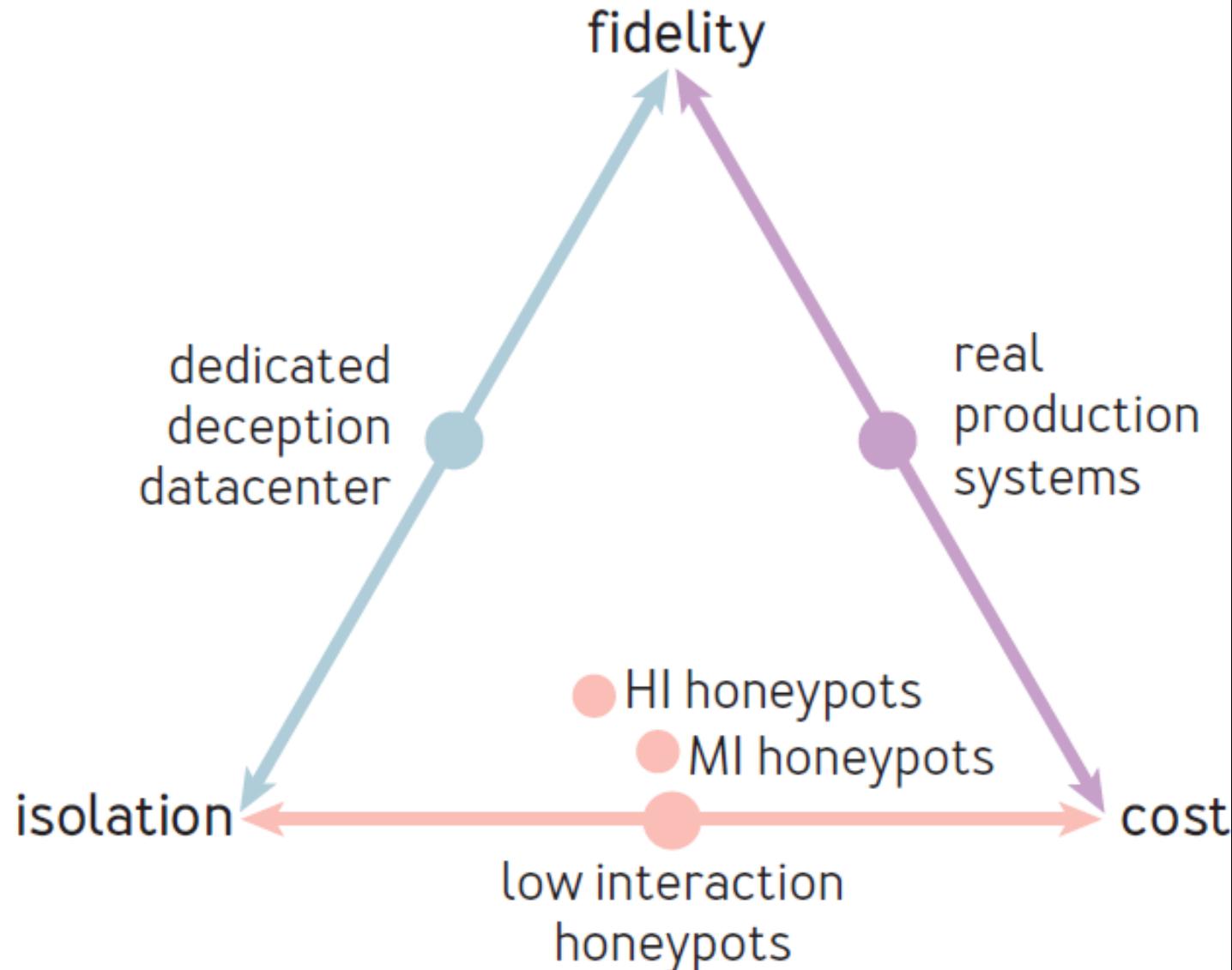


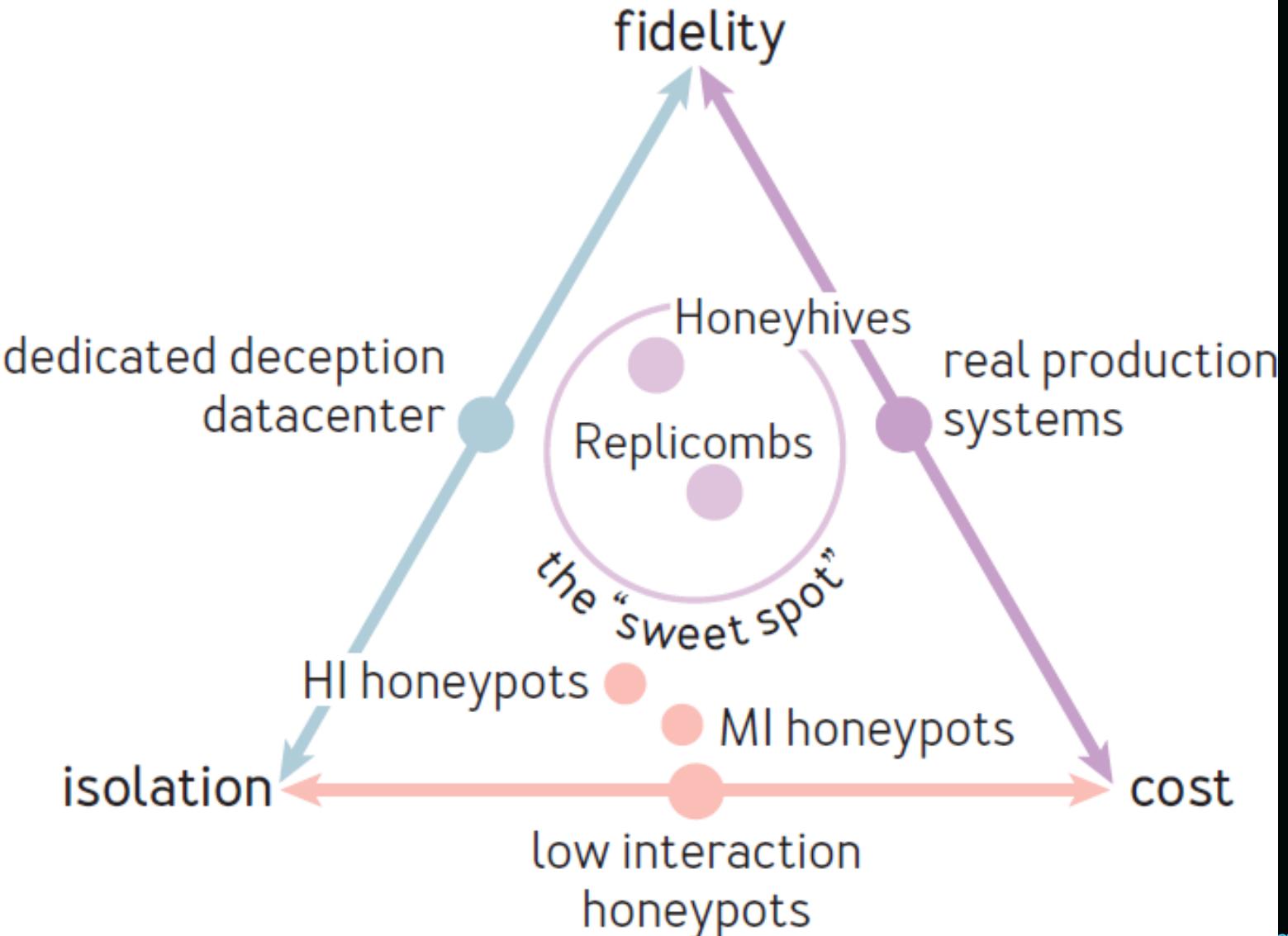
FIGURE 3: MI & HI HONEYPOTS ON THE TRILEMMA



A photograph of two heart-shaped lollipops against a teal background. One lollipop is pink and the other is blue. They are positioned in the lower half of the frame, with the pink one on the left and the blue one on the right.

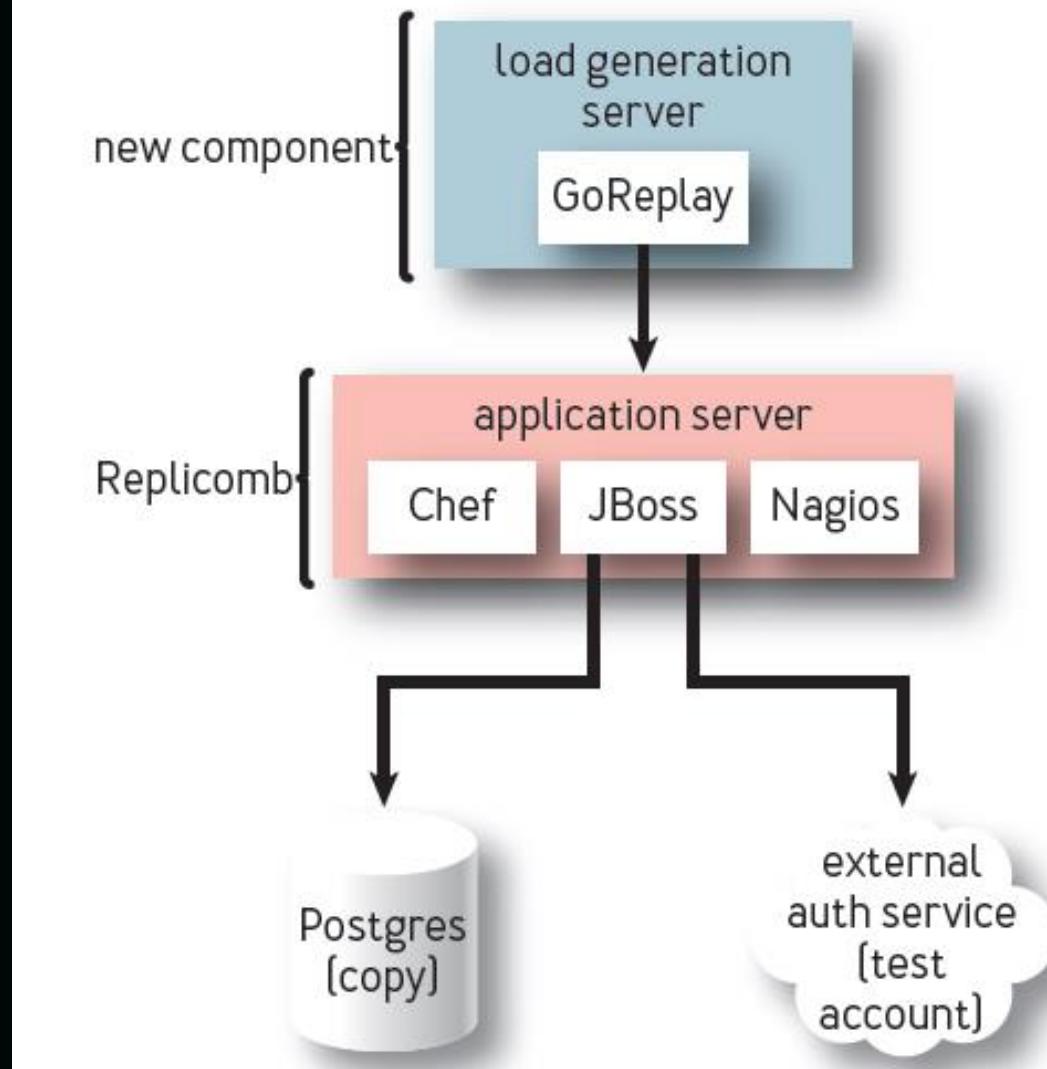
But there is a previously unexplored “sweet spot” for deception mechanisms...

FIGURE 4: THE FIC “SWEET SPOT”: HONEYHIVES AND REPLICOMBS



Replicombs: downgraded replicas of prod hosts with the same services seen in prod

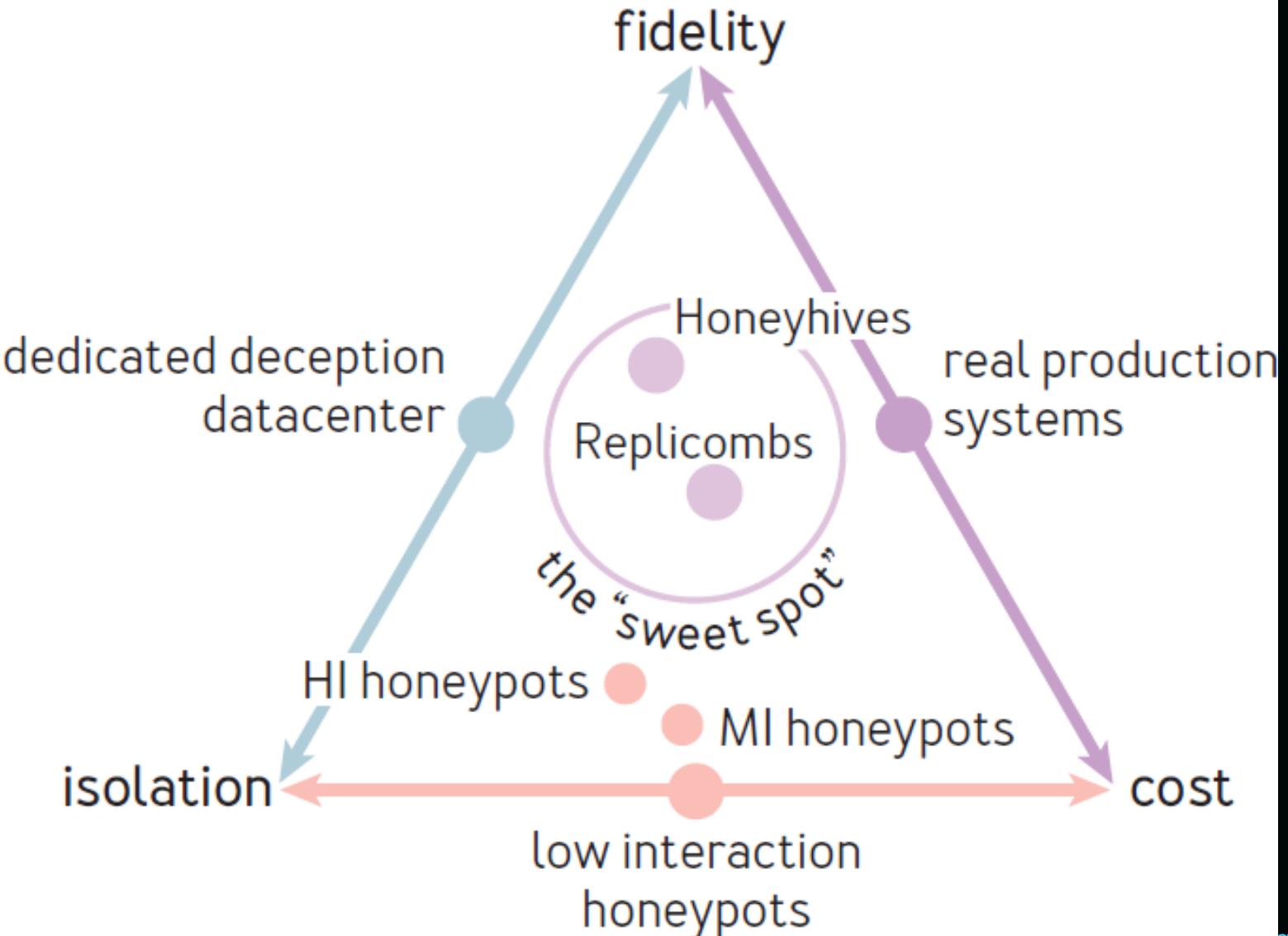
FIGURE 5: AN EXAMPLE REPLICOMB DEPLOYMENT



A close-up photograph of a donut with pink frosting and colorful sprinkles. The sprinkles are various shapes and colors, including stars, dots, and crosses in shades of blue, yellow, green, and white. The donut is positioned in the upper right quadrant of the frame, with a blurred background.

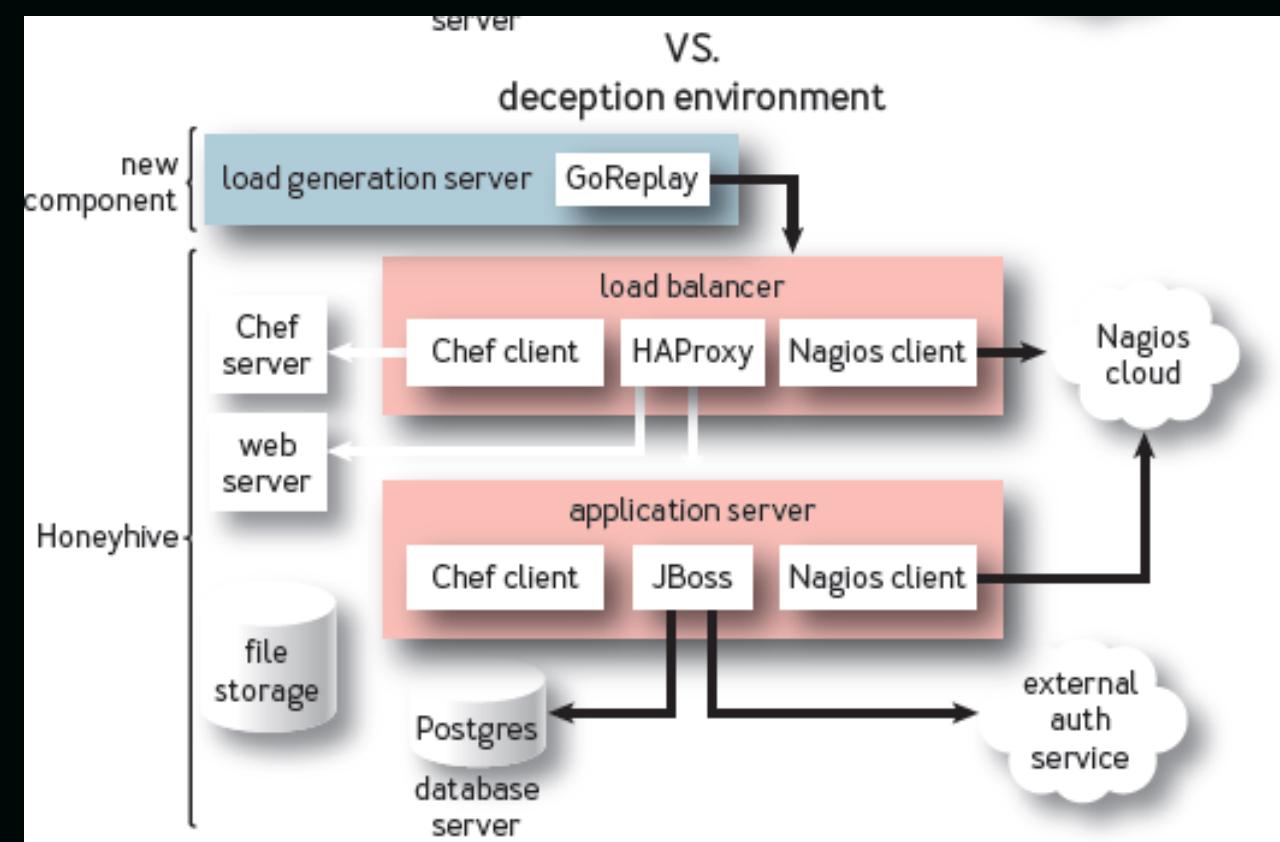
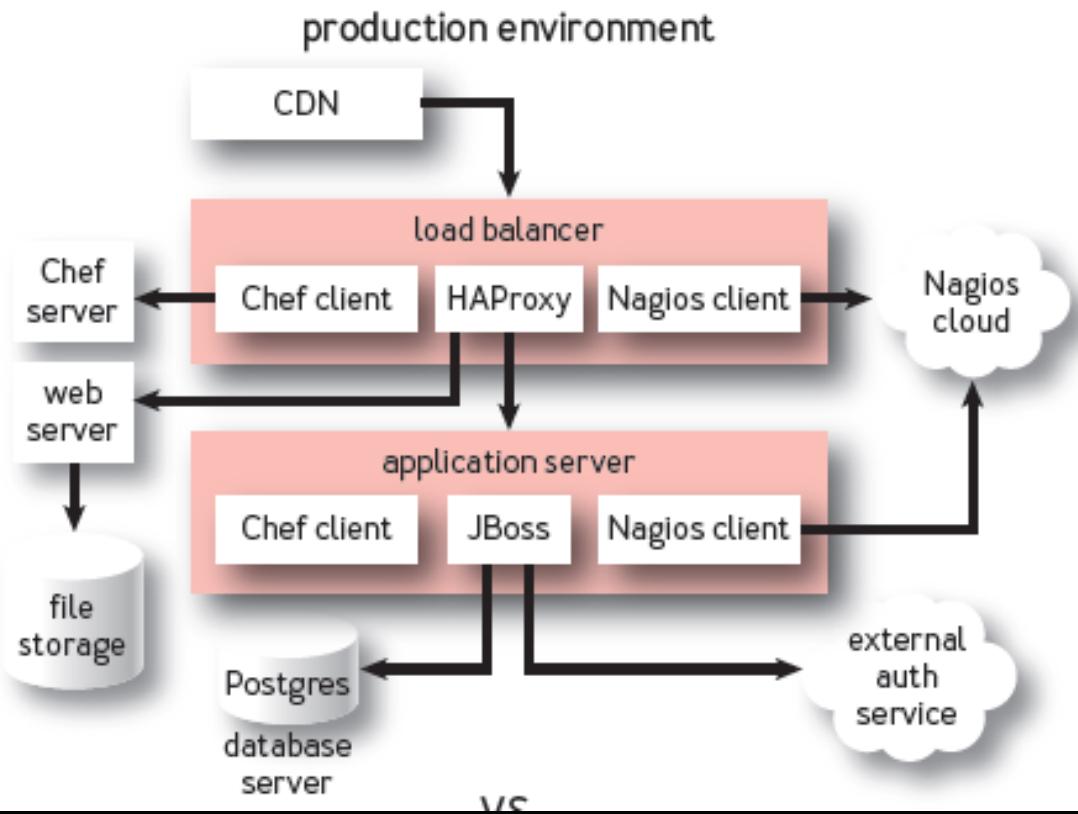
Replicomb vs. honeypot: impressive fidelity
with an expansive range of attack behavior

FIGURE 4: THE FIC “SWEET SPOT”: HONEYHIVES AND REPLICOMBS

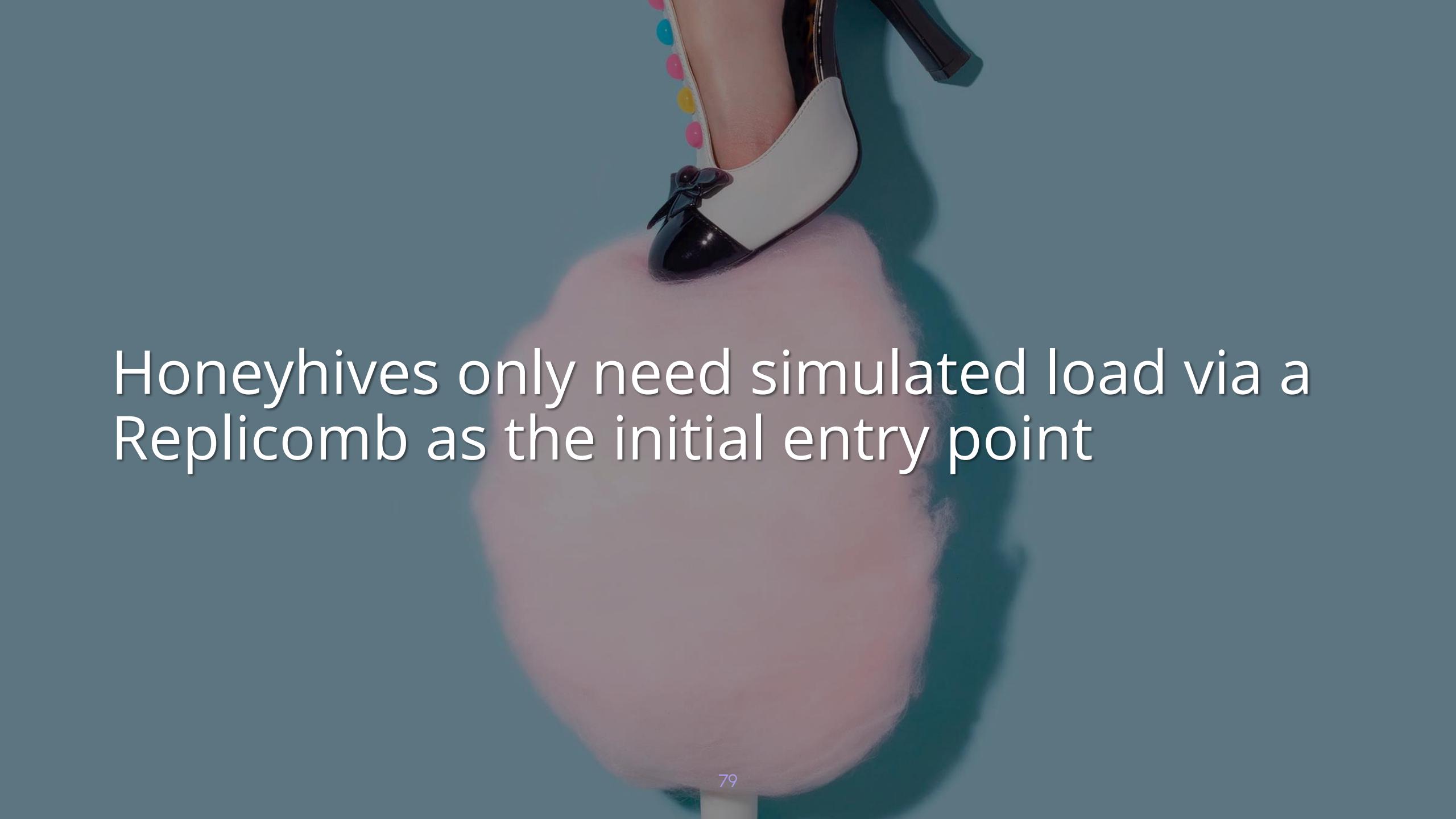


Honeyhives: full network of like-prod hosts
to observe attacker movement x-system

FIGURE 6: EXAMPLE HONEYHIVE BASED ON A PRODUCTION ENVIRONMENT



Modern IaC practices + inexpensive full isolation via cloud computing are key

A close-up photograph of a person's hand holding a large, light-colored honeycomb. A small, dark-colored bee is visible on the surface of the honeycomb. The person is wearing a black high-heeled shoe with a white toe cap and a decorative band featuring colorful circles (pink, yellow, blue) around the ankle. The background is a solid teal color.

Honeyhives only need simulated load via a
Replicomb as the initial entry point

Okay, but how tf do you implement this in
the real world of messy software eng?

Actually, it's no more difficult than setting up a new variant of an existing env tier...



Replicomb is similar to a canary release.
Honeyhive is like a soak or load test env.

But there are details to consider when implementing this in your org...



Isolation boundaries: DEs need to be properly isolated from user traffic

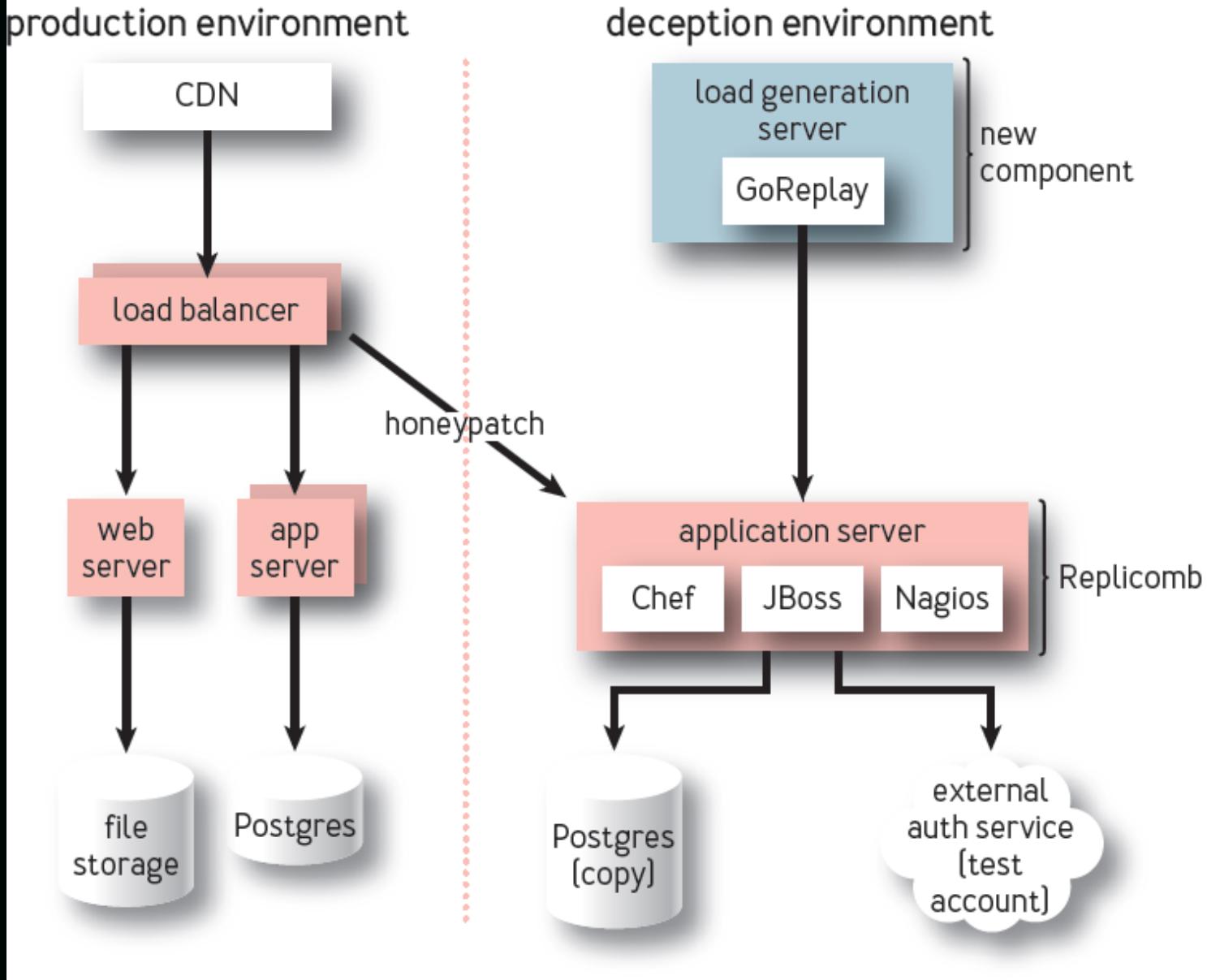
Virtualization, SDNs, cloud computing can help create fully isolated networks for DEs

A stack of colorful macarons (yellow, purple, blue, pink) on a light surface. A dark gray rectangular overlay covers the top half of the image, containing white text.

Discoverability: attackers need to find the DE for you to collect real data on their ops

Honeypatching can support discoverability

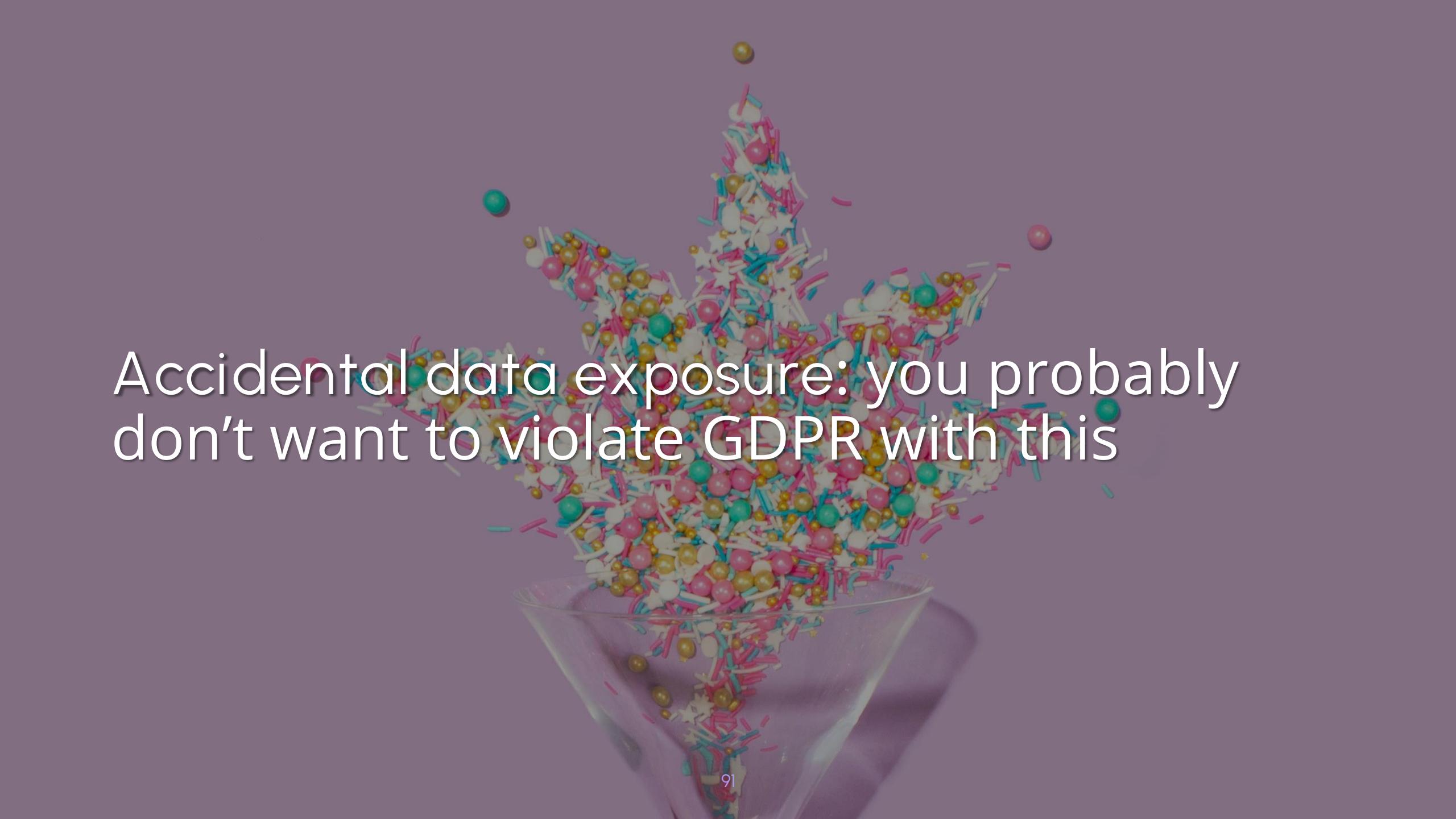
FIGURE 7: AN EXAMPLE REPLICOMB ENVIRONMENT WITH HONEYPATCHING



A row of colorful milkshakes with straws and sprinkles.

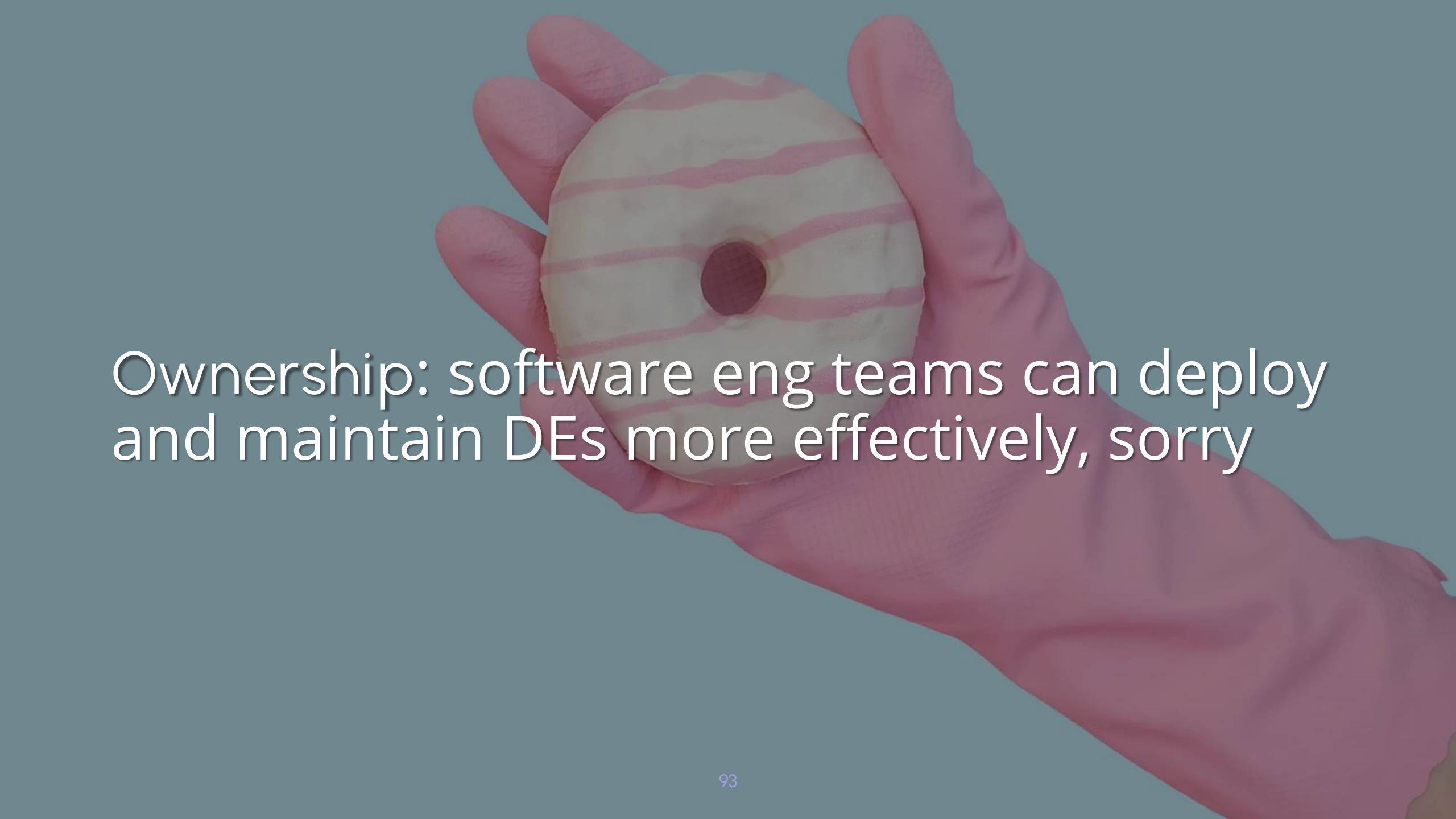
Tamper-free observation: tracing should be
invisible to attackers + resistant to tamper

Traffic archiving, memory & disk snapshots,
process launch events, file activity...

A martini glass is filled to the brim with a variety of colorful candy sprinkles, including stars, dots, and larger spheres in shades of pink, blue, yellow, and white. The glass is tilted, causing the sprinkles to spill out onto a light-colored surface. The background is a solid dark purple.

Accidental data exposure: you probably
don't want to violate GDPR with this

Mitigation: anonymize or scramble traffic
or generate synthetic data sets to replay

A close-up photograph of a person's hand wearing a light blue glove. The hand is holding a white frosted donut with several horizontal pink stripes and a single hole in the center. The background is a solid teal color.

Ownership: software eng teams can deploy
and maintain DEs more effectively, sorry

SWEs can treat attackers as a kindred engineer with the exact opposite goals

V. Harvesting potential

1. Resilient system design
2. Attacker tracing
3. Experimentation platform

Resilient system design



DEs let you explore how attacks impact systems to inform design improvements

Attackers interact with monitoring, logging, alerting, failover, and service components in ways that stress their overall reliability

A photograph of a beach at sunset. The sky is a warm orange and yellow, transitioning into a darker blue. The ocean waves are crashing onto the sandy shore, creating white foam. The sand is dark and textured. The overall atmosphere is peaceful and scenic.

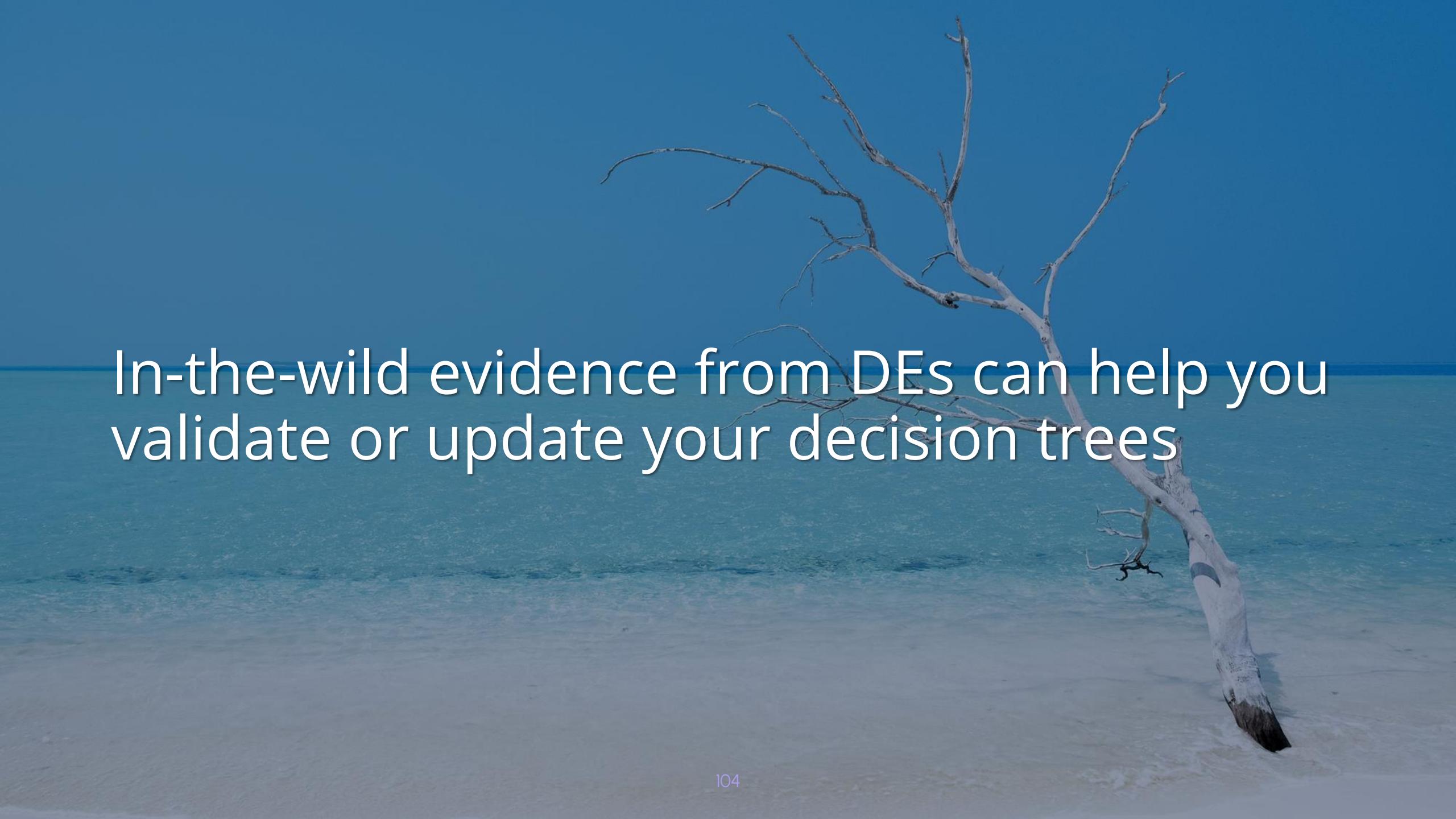
DEs expose opportunities for architectural
improvements in operability & simplicity

Eng teams can leverage a feedback loop
fueled by real-world evidence from DEs

The background of the image shows a close-up view of a sandy ocean floor. The sand is light-colored and has a fine, granular texture. Small, silvery fish are scattered across the surface, some near the bottom and others slightly above it. The water is clear enough to see the sandy bottom, and the overall scene is a natural, underwater environment.

Attacker tracing

Attack observability enables pragmatic threat modeling during design & planning

A photograph of a dead tree standing in shallow, clear blue water. The tree has many bare, twisted branches reaching outwards. The background is a bright, clear blue sky.

In-the-wild evidence from DEs can help you validate or update your decision trees

Text Editor

```

title: (Example) Attack Tree for S3 Bucket with Video Recordings

facts:
- wayback: API cache (e.g. Wayback Machine)
  from:
  - reality: '#yolosec'
- public_bucket: S3 bucket set to public
  from:
  - bucket_search: '#yolosec'
- subsystem_with_access: Subsystem with access to bucket data
  from:
  - compromise_user_creds

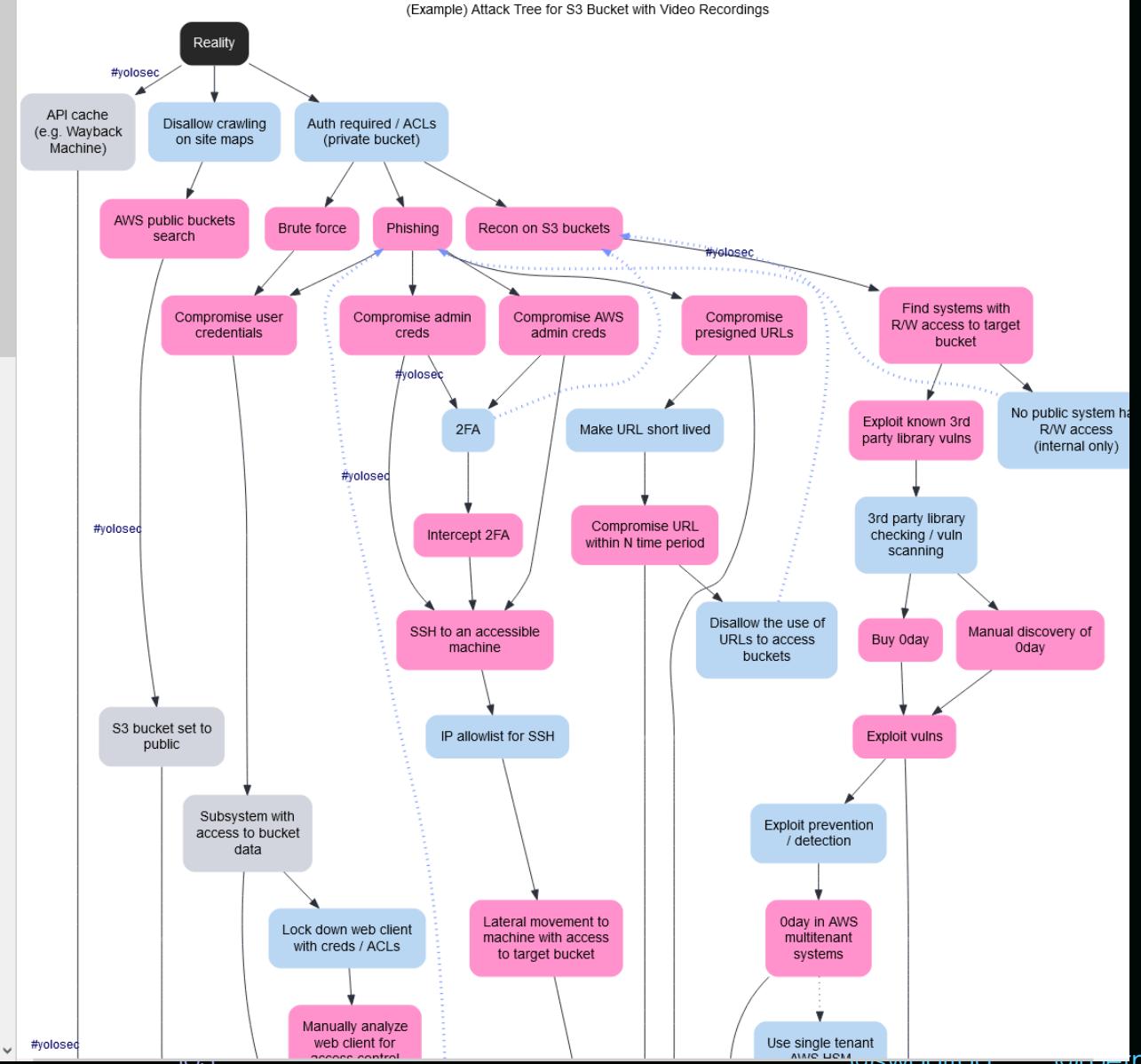
attacks:
- bucket_search: AWS public buckets search
  from:
  - disallow_crawling
- brute_force:
  from:
  - private_bucket
- phishing:
  from:
  - private_bucket
  - internal_only_bucket:
    backwards: true
  - access_control_server_side:
    backwards: true
- compromise_user_creds: Compromise user credentials
  from:
  - brute_force
  - phishing
- analyze_web_client: Manually analyze web client for access control misconfig
  from:
  - lock_down_acls
- compromise_admin_creds: Compromise admin creds
  from:
  - phishing
- compromise_aws_creds: Compromise AWS admin creds
  from:
  - phishing
- intercept_2fa: Intercept 2FA
  from:
  - 2fa
- ssh_to_public_machine: SSH to an accessible machine
  from:
  - compromise_admin_creds: '#yolosec'
  - compromise_aws_creds:
  - intercept_2fa
- lateral_movement_to_machine_with_access: Lateral movement to machine with access to
  from:
  - ip_allowlist_for_ssh
- compromise_presigned: Compromise presigned URLs
  from:
  - phishing
- compromise_quickly: Compromise URL within N time period
  from:
  - ...

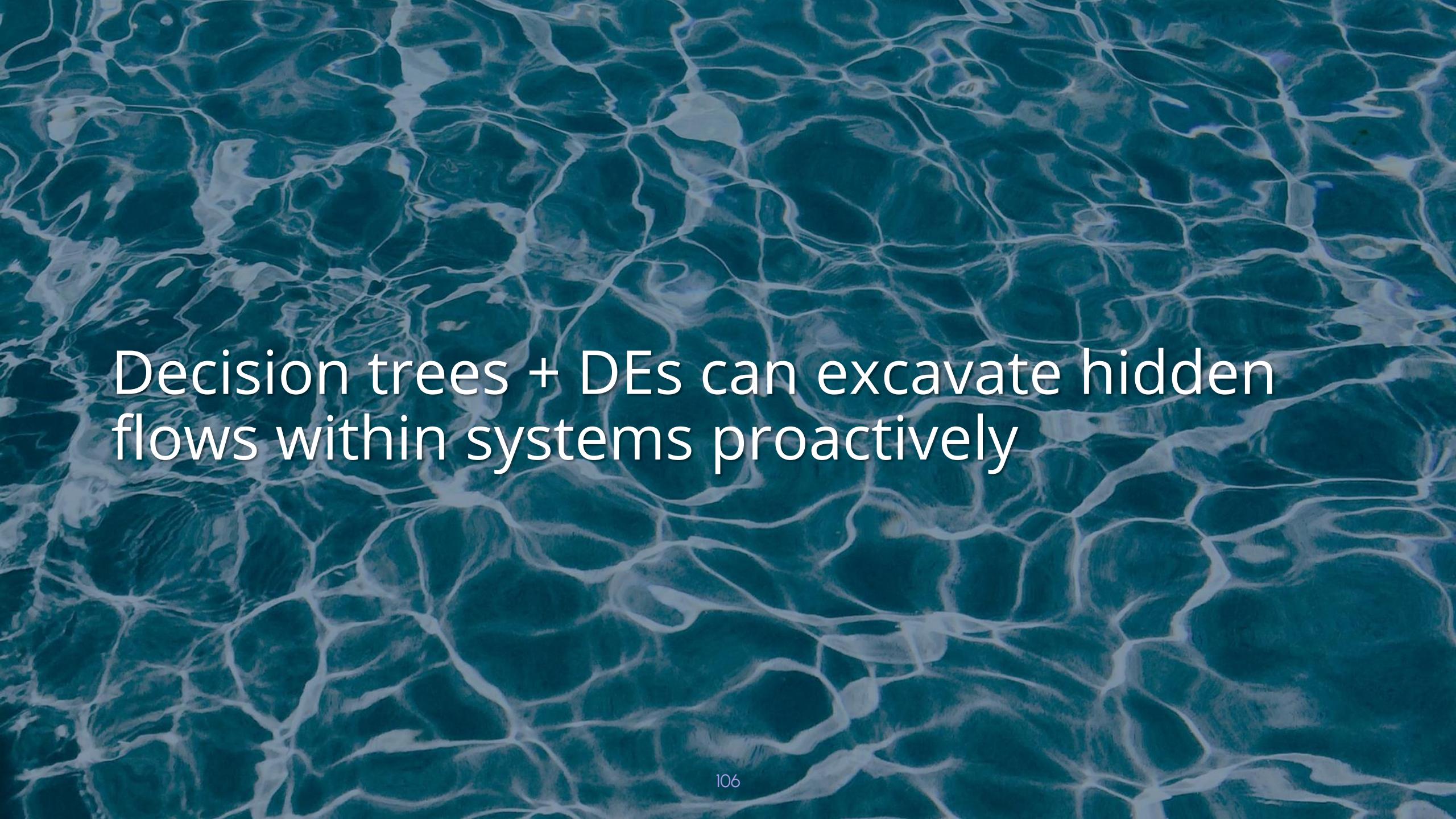
```



Inspired by and with example taken from Kelly Shortridge's [Creating Security Decision Trees With Graphviz](#)

[Import GitHub Gist](#) [Download .svg](#) [Download .dot](#)





Decision trees + DEs can excavate hidden flows within systems proactively

Attacker tracing also fuels experimentation:
each branch is a chain of hypotheses

Experimentation platform

Experimentation can test the efficacy of monitoring or resilience measures

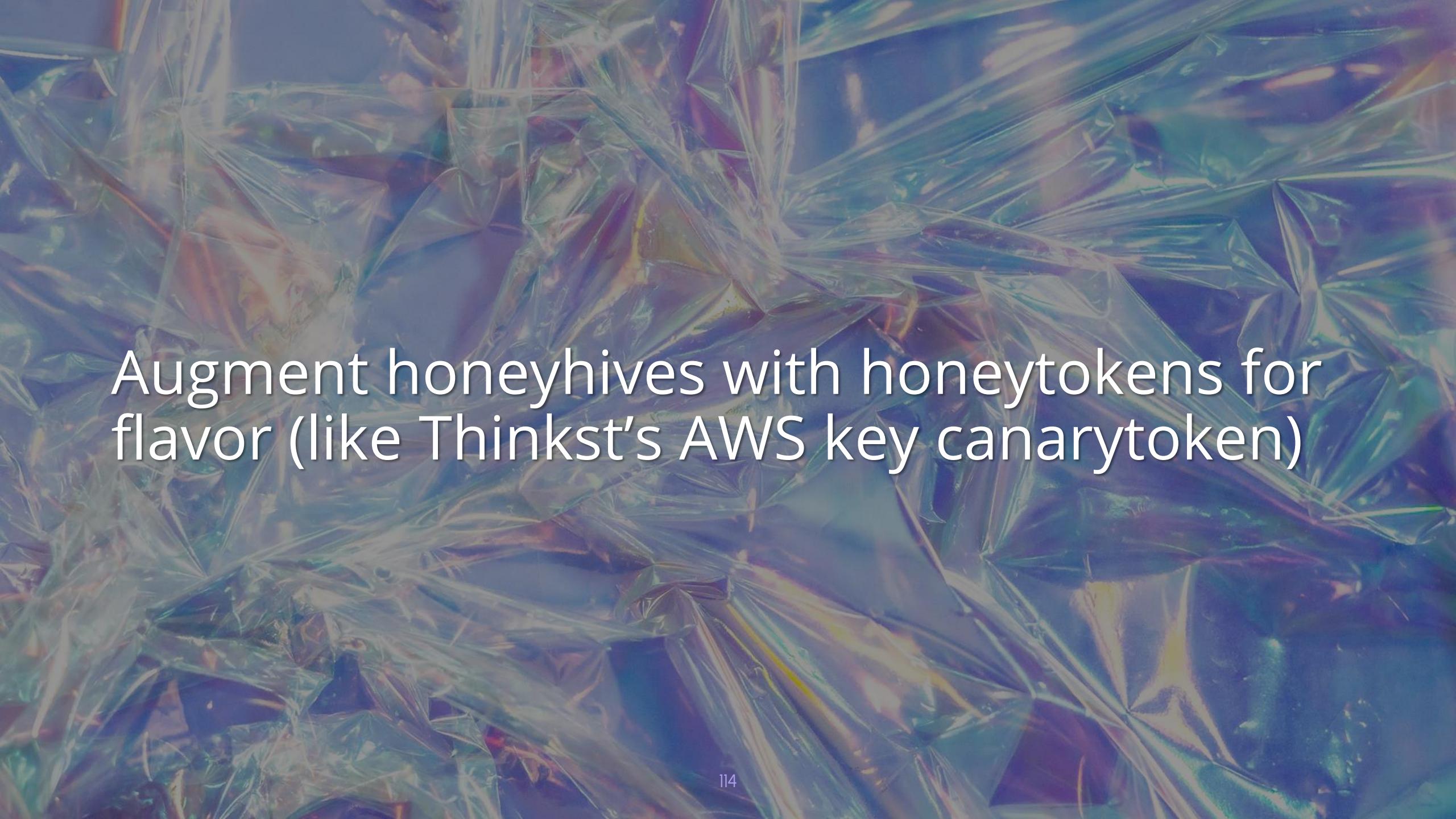


Deception Environments become a tool in
the Security Chaos Engineering arsenal

Fidelity degradation experiments divulge
how attackers react to different envs

Swap standard components for substitutes
to disrupt attack plans in prod (sow F.U.D.)

Tune the difficulty of accessing the DE to study different types of attackers

The background of the slide features a complex, abstract pattern of translucent, multi-colored shapes resembling shards of glass or crystalline structures. The colors range from deep blues and purples to bright yellows and reds, creating a sense of depth and light refraction.

Augment honeyhives with honeytokens for flavor (like Thinkst's AWS key canarytoken)

VI. Future opportunities

The background of the slide is a dark blue space scene. It features numerous small white stars of varying sizes scattered across the frame. In the center, there is a faint, glowing nebula or galaxy with a mix of blue, purple, and yellowish hues, appearing as a soft, hazy cloud.

Just-in-time terraforming

JIT creation of isolated deception VMs via
copy-on-write or page deduplication



Systems terraforming: reify an entire constellation of hosts upon connection

Potential network & hypervisor tricks:
unfreeze assets & fast-forward execution...

Virtualization is one big lie to software—
why not extend this lie a little bit further?



Instance emulation

Full emulation of CSP APIs would facilitate
DEs but also other operational benefits...

Honeypatching at scale: redirect attackers towards a DE + deploy via update pipelines



Anonymization via mirroring

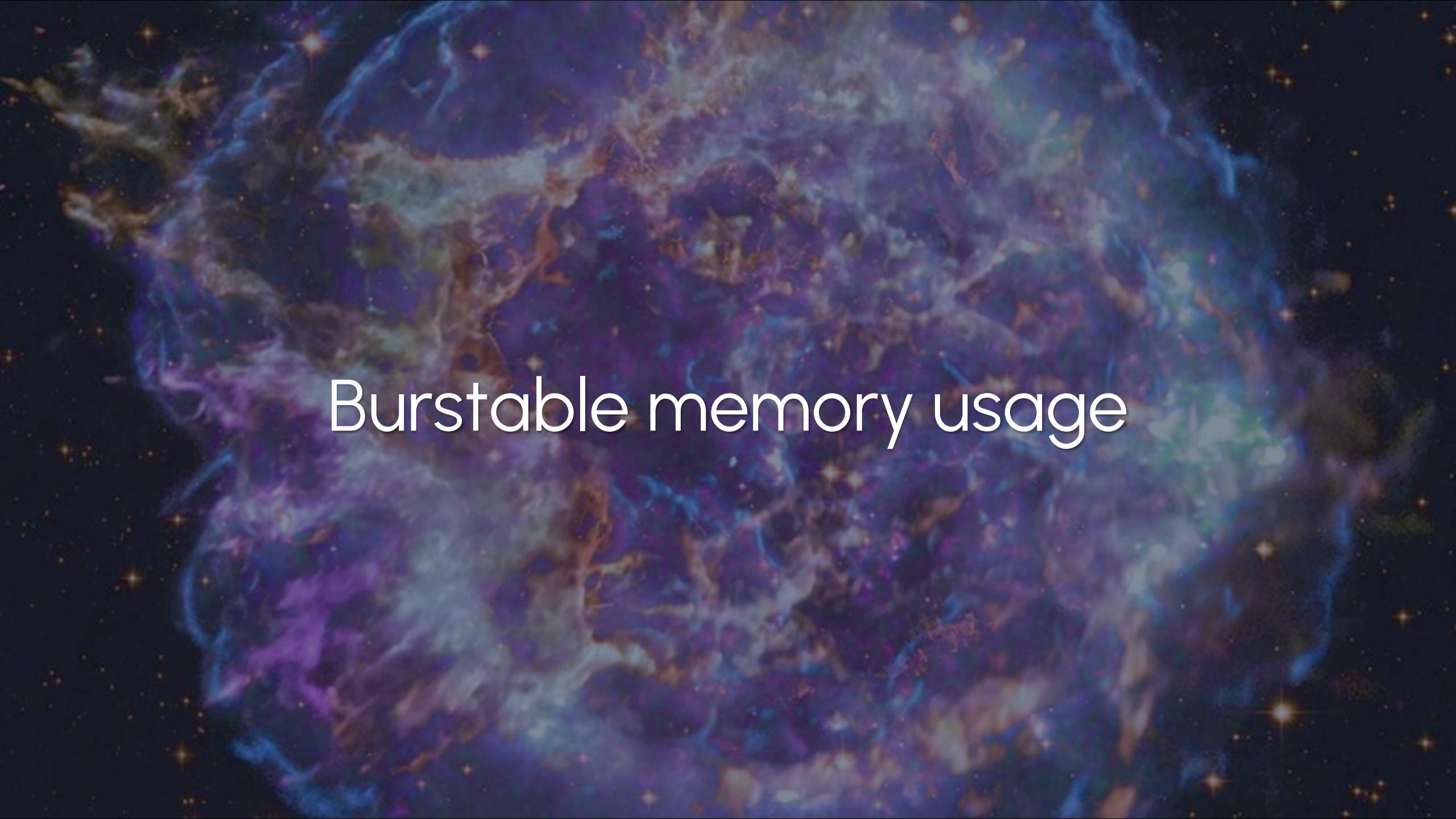
Extend traffic-mirroring tech to include
data anonymization features (layer 7 ftw)



Hypervisor-based observability

Tracing & observability tools often execute with root privileges & are simple to subvert

OSes could expose core events (process and file ops) over a common protocol...

The background of the image is a deep, dark space filled with numerous small, glowing stars of varying sizes. Overlaid on this are several large, luminous nebulae. One prominent nebula in the center-right is a dense cluster of blue and purple light, with wispy extensions of the same color extending towards the top left and bottom right. Another smaller, more diffuse nebula is visible in the lower-left quadrant. The overall effect is one of a dynamic, energetic, and infinite universe.

Burstable memory usage

CSPs could support burstable performance instances via ballooning or swapped mem

Temporarily migrate VMs across physical instances when their activity bursts...



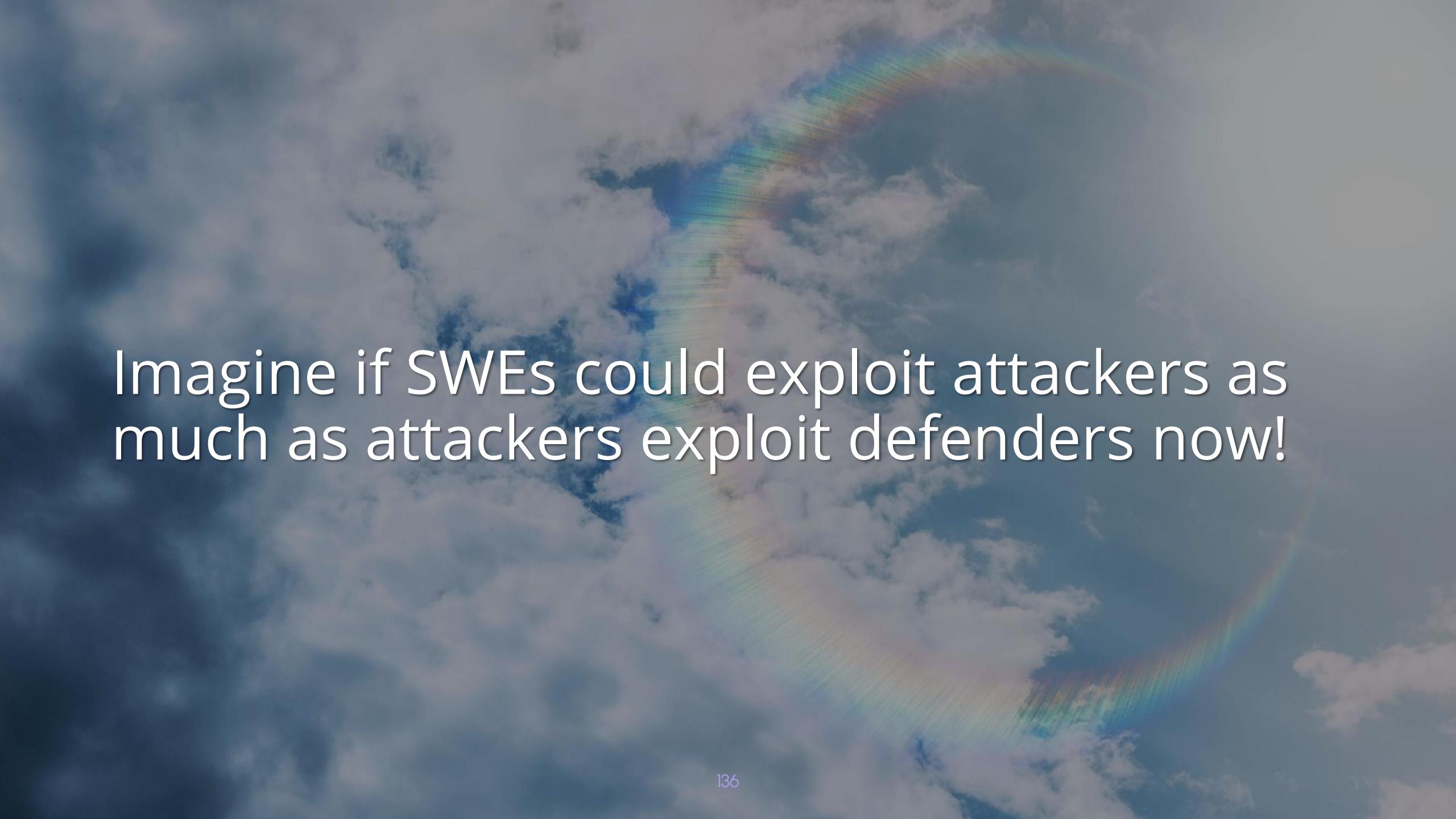
Per-account billing limits

Per-account billing limits can restrict the amount of your \$\$\$ attackers can spend

CSPs have effective tools for isolation every resource except for customers' wallets

A wide-angle photograph of a vibrant rainbow arching over a vast, choppy ocean under a cloudy sky.

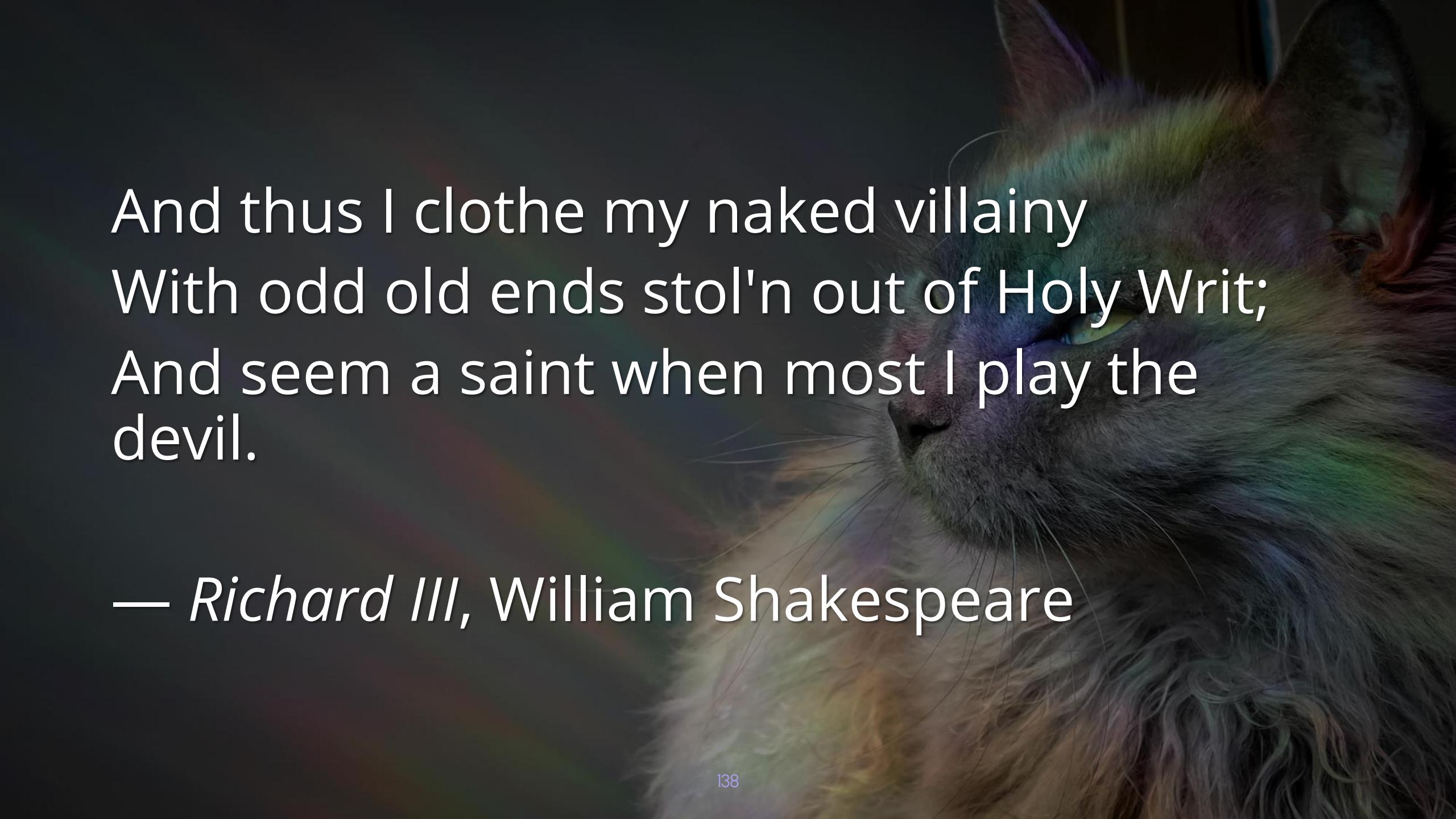
VI. Conclusion

A photograph of a bright rainbow arching across a sky filled with white and grey clouds. The rainbow's colors are vivid, transitioning from red at the top to blue at the bottom. The background is a mix of light and dark clouds.

Imagine if SWEs could exploit attackers as
much as attackers exploit defenders now!

The background of the slide features a close-up photograph of several green palm fronds in the foreground, angled upwards towards the top left. Behind them is a vast, hazy sky at sunset or sunrise, transitioning from a deep blue-grey at the top to warm orange and yellow hues near the horizon. A faint, multi-colored rainbow arc is visible in the upper right quadrant.

Deception envs allow you to bamboozle
attackers for fun and profit (and resilience)



And thus I clothe my naked villainy
With odd old ends stol'n out of Holy Writ;
And seem a saint when most I play the
devil.

— *Richard III*, William Shakespeare



@swagitda_



@rpetrich



/in/kellyshortridge



/in/rpetrich



chat@shortridge.io



rpetrich@gmail.com