



# LAMBOOZLING ATTACKERS

Kelly Shortridge (@swagitda\_)

Ryan Petrich (@rpetrich)

Summercon 2022



A woman with blonde hair, wearing sunglasses and a black bikini top, is standing in the ocean waves. She is holding a book titled "Security Chaos Engineering" by Aaron Brethart & Kelly Sherrill. She has her tongue sticking out and her arms outstretched, appearing to be having fun. The background is the blue ocean with white foam from the waves.

Hi, I'm Kelly  
**fastly**





A man with glasses and a watch is sitting on a rocky beach. He is shirtless and wearing dark shorts. He is smiling and looking towards the camera. The ocean is behind him, with waves crashing against the rocks. The sky is blue and clear. The text "Hi, I'm Ryan" is overlaid on the image in a white, sans-serif font.

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 person is floating in the ocean on a large, pink, inflatable ring. The person is wearing a light-colored swimsuit and a straw hat. The water is a deep blue with white foam from the waves. The overall scene is a high-angle shot of the ocean.

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



A close-up photograph of a green palm frond, showing the individual leaflets, set against a solid, muted purple background. The frond is positioned diagonally across the frame, with its base on the left and its tip extending towards the upper right.

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



The background of the slide is a deep blue, textured surface, possibly representing a planet or a vast body of water. A vibrant rainbow arches across the top half of the image, its colors blending into the blue background. The text is centered in the lower half of the image.

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



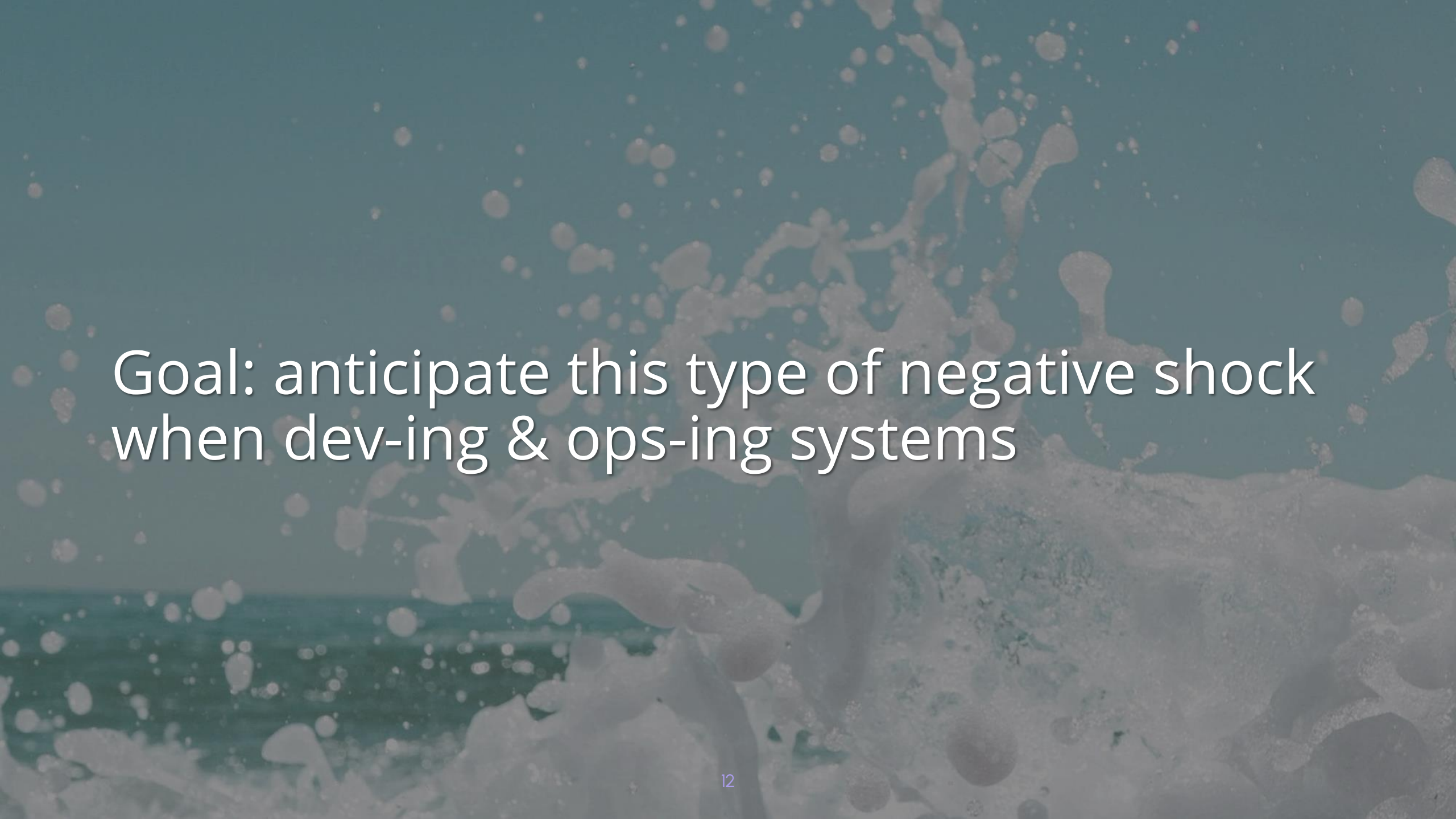


# I. Exploiting attacker brains (for fun & profit)



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

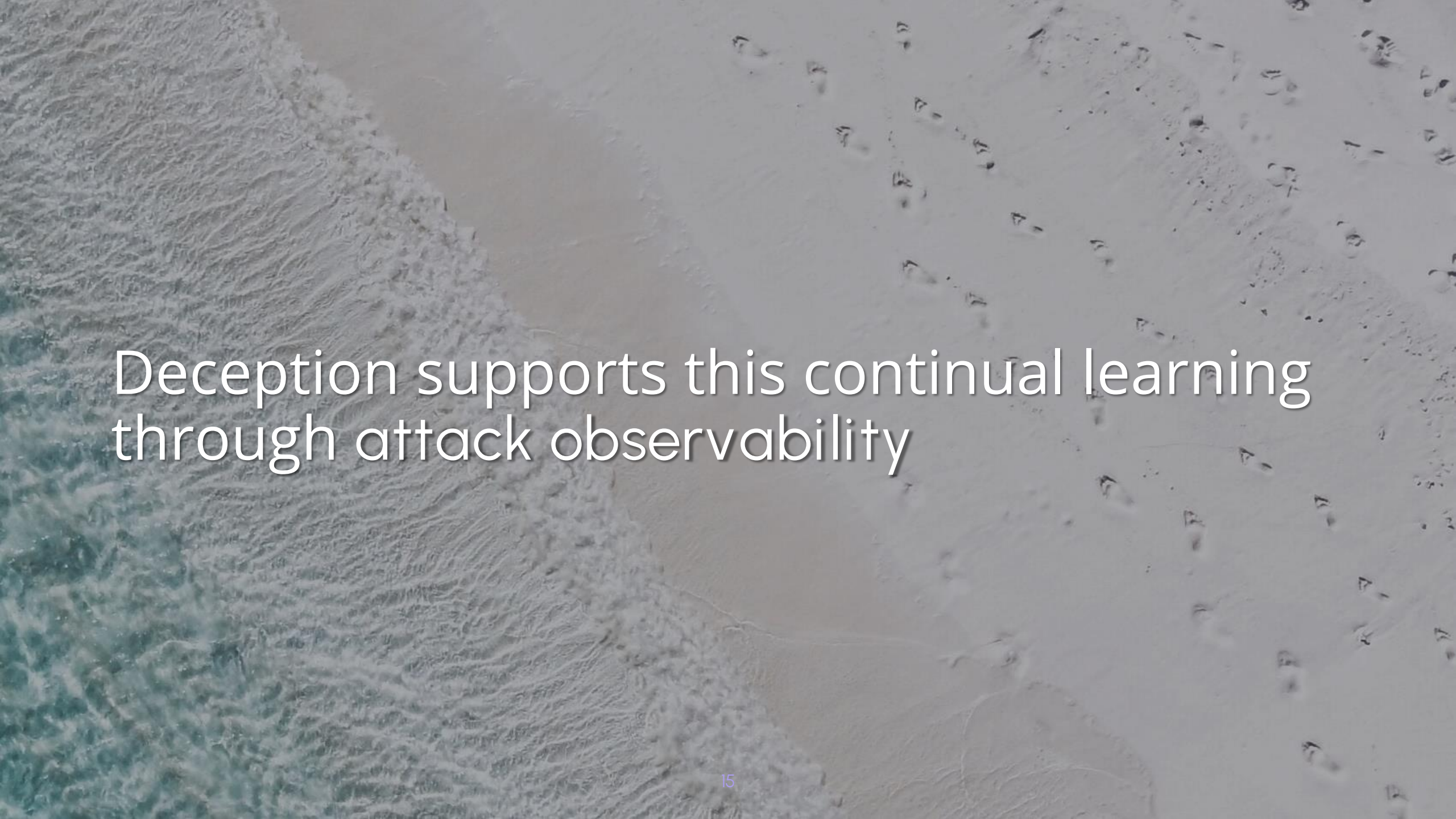


1. Collect relevant info about attackers
2. Implement anticipatory mechanisms that impede the success of attack ops



Sustaining resilience in complex systems  
requires a continual learning capacity



An aerial photograph of a beach. On the left, dark, turbulent waves are crashing onto the shore. The sand is a light tan color. Numerous footprints are visible in the sand, mostly on the right side of the frame, leading away from the water. The text "Deception supports this continual learning through attack observability" is overlaid in white with a thin black outline, centered horizontally and slightly above the middle vertically.

Deception supports this continual learning  
through attack observability



Attack Observability: observing the interaction between attackers & systems



A tropical scene with palm trees and a rainbow. The background is a clear blue sky with a vibrant rainbow arching across it. Several palm trees are visible, with their fronds reaching towards the top of the frame. The text is overlaid on the left side of the image.

Actual system behavior in production  
notoriously deviates from expectations



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



An aerial photograph of a tropical beach. The top half of the image shows clear, turquoise ocean water with visible ripples and some white foam from a wave breaking near the shore. The bottom half shows a wide, sandy beach. A single palm tree stands prominently on the right side of the beach, its shadow cast long and dark onto the sand. Other palm trees are visible in the lower corners, and their shadows are also cast on the sand. The overall scene is serene and idyllic.

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

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



The background of the slide features two palm trees silhouetted against a sky that transitions from a deep blue at the top to a soft pink at the bottom. The text is centered horizontally and partially overlaps the palm trees.

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 serene tropical beach scene at sunset or sunrise. A tall palm tree stands on the left, its fronds reaching towards a clear, light blue sky. The calm water of the ocean reflects the tree and the sky, creating a mirror-like effect. The horizon line is visible in the distance, with a few small boats on the water. The overall atmosphere is peaceful and idyllic.

Operators can use deception to amplify  
information asymmetries in their favor

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



A low-angle photograph of palm trees against a sunset sky. The sky is a mix of soft pinks, purples, and blues, with a faint rainbow visible in the background. The palm trees are dark silhouettes, with their fronds reaching upwards. The text is overlaid in the center of the image.

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)

The background of the slide is a close-up photograph of numerous water droplets of various sizes. The droplets are clear and spherical, reflecting light in a way that creates bright highlights and darker shadows, giving them a three-dimensional appearance. They are scattered across a light-colored, slightly textured surface, possibly a piece of paper or a clean glass pane. The overall color palette is soft and natural, dominated by the whites and grays of the water and the subtle textures of the surface.

## II. 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?

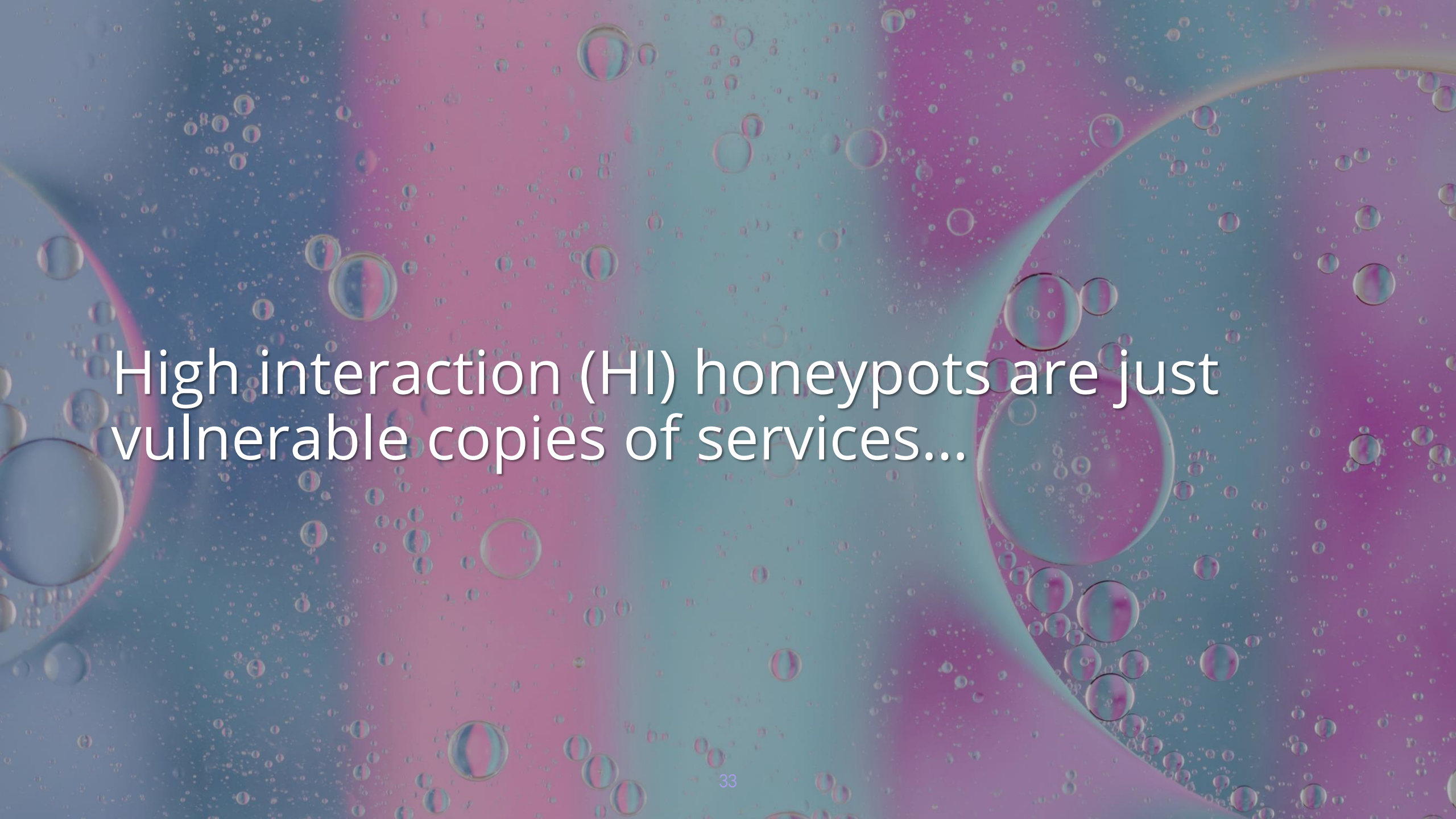




Low interaction (LI) honeypots are basically cardboard-cutout decoys...

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





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

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



The background of the slide is a soft blue gradient with numerous out-of-focus water bubbles of various sizes. A cluster of larger, more defined bubbles is positioned behind the text, adding a sense of depth and texture.

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

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



A close-up photograph of a slice of orange being dropped into water. The orange slice is partially submerged, with its bright yellow-orange interior and white pith visible. A large, dense cloud of fine, white bubbles erupts from the point of impact, filling the upper half of the frame. The water is a clear, light blue color. The overall scene conveys a sense of freshness and dynamic movement.

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





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?



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




Deployment automation + defining infra declaratively decreases ops overhead

The background of the slide is a gradient of blue and purple, with three birds in flight. One bird is in the center, slightly above the text, with its wings spread. Another bird is to the left, and a third is to the right, both also in flight. The text is white with a slight shadow, making it stand out against the background.

Virtualization advancements: isolation,  
observability, denser computing



SDN proliferation enables isolated network topology dedicated to attackers



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

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



The background of the slide is a full-page image of marbled paper. It features intricate, swirling patterns in a variety of colors including deep blues, purples, pinks, and greens, creating a complex and organic texture.

# IV. Designing deception environments

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




The background of the slide features a repeating pattern of stylized, colorful swirls on sticks, resembling lollipops or candy canes. The swirls are composed of concentric rings in shades of pink, blue, and yellow, set against a solid brown background. The sticks are thin and light-colored, extending from the bottom of each swirl.

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”

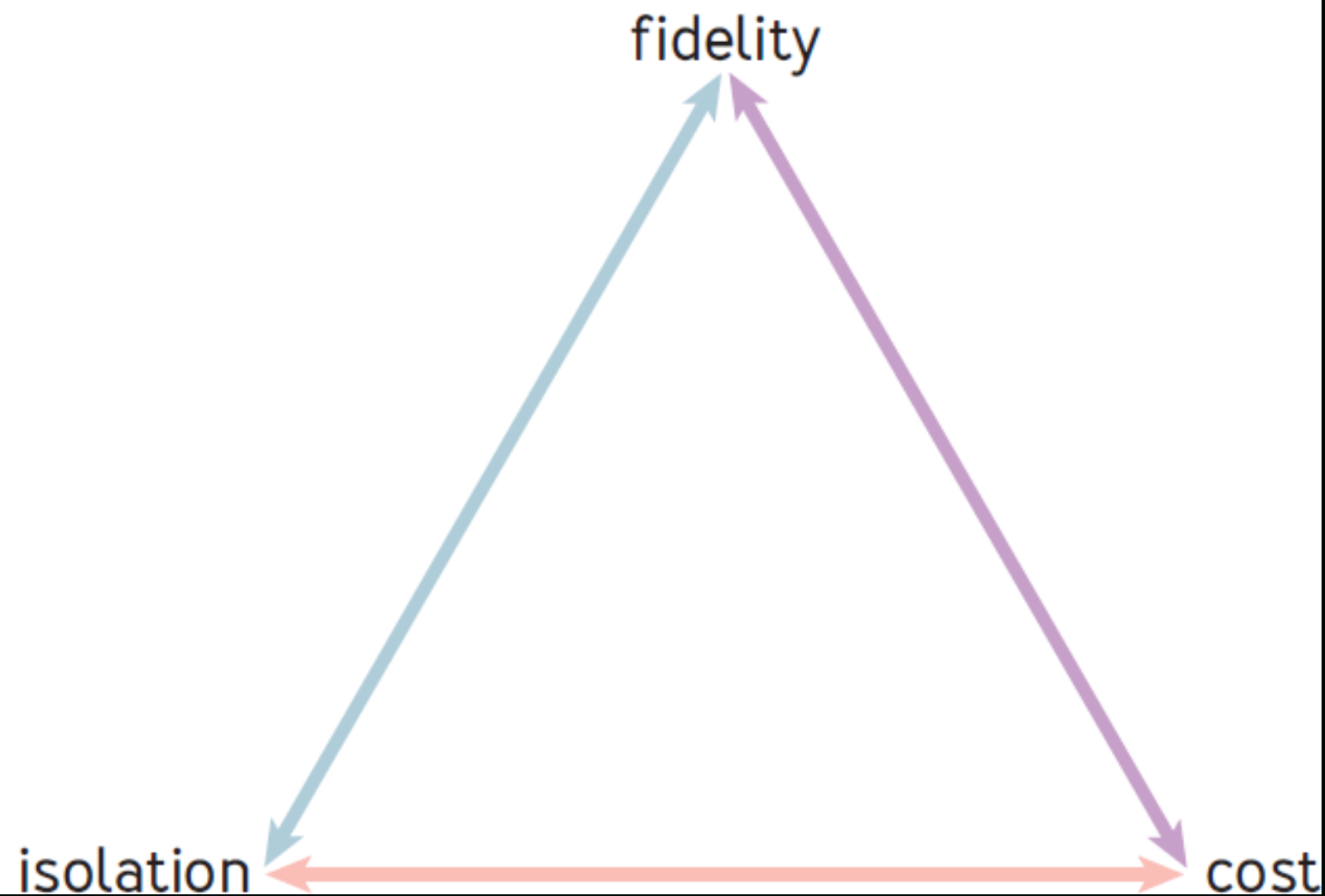
A background image featuring several macarons in various colors (green, pink, yellow, orange) scattered around the edges of a solid pink background. One pink macaron in the upper center is broken, showing a dark filling.

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 cake, likely a birthday cake, featuring white frosting with pink and yellow swirls. The cake is decorated with colorful sprinkles, including pink, yellow, and blue. The background is a soft, out-of-focus grey.

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



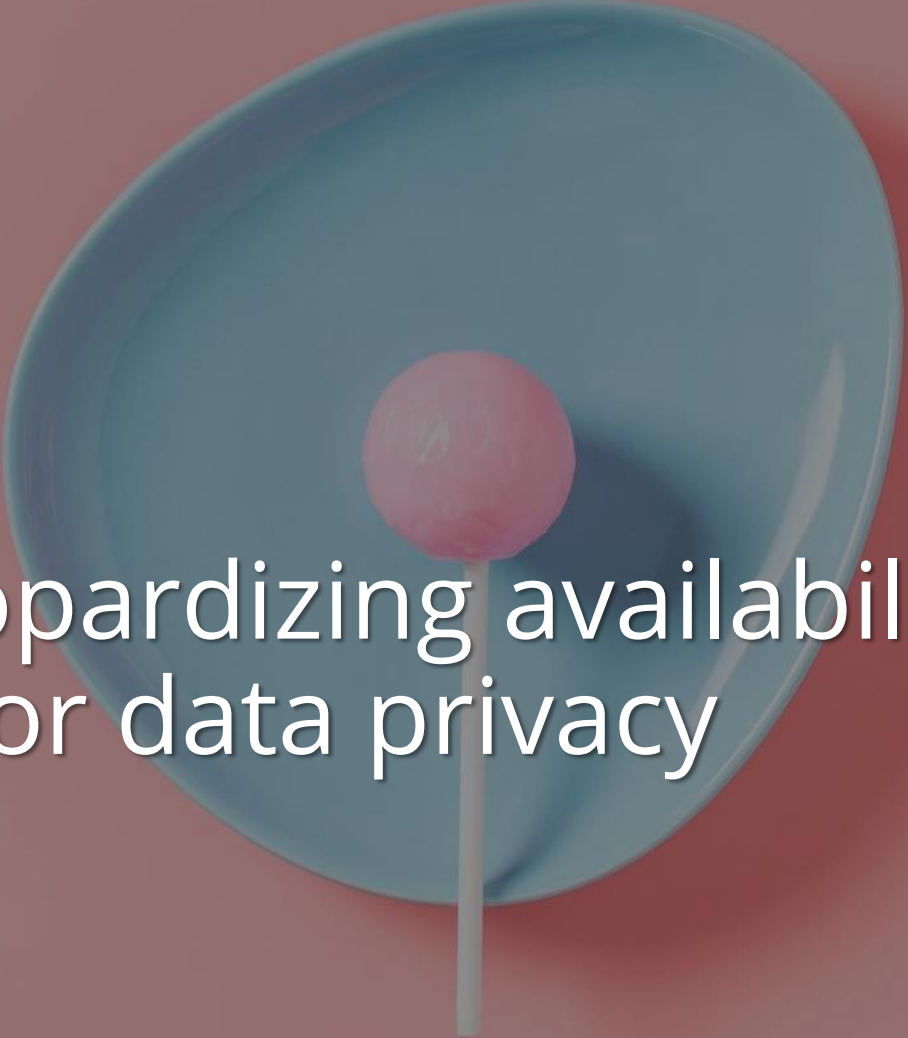
Deception systems need sufficient depth to stimulate extended attacker activity



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



A blue funnel is centered in the upper half of the image. Inside the narrow neck of the funnel is a pink lollipop. The background is a solid red color. The text "Goal: not jeopardizing availability of the real system or data privacy" is written in white, sans-serif font across the middle of the image, partially overlapping the funnel and the lollipop.

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**

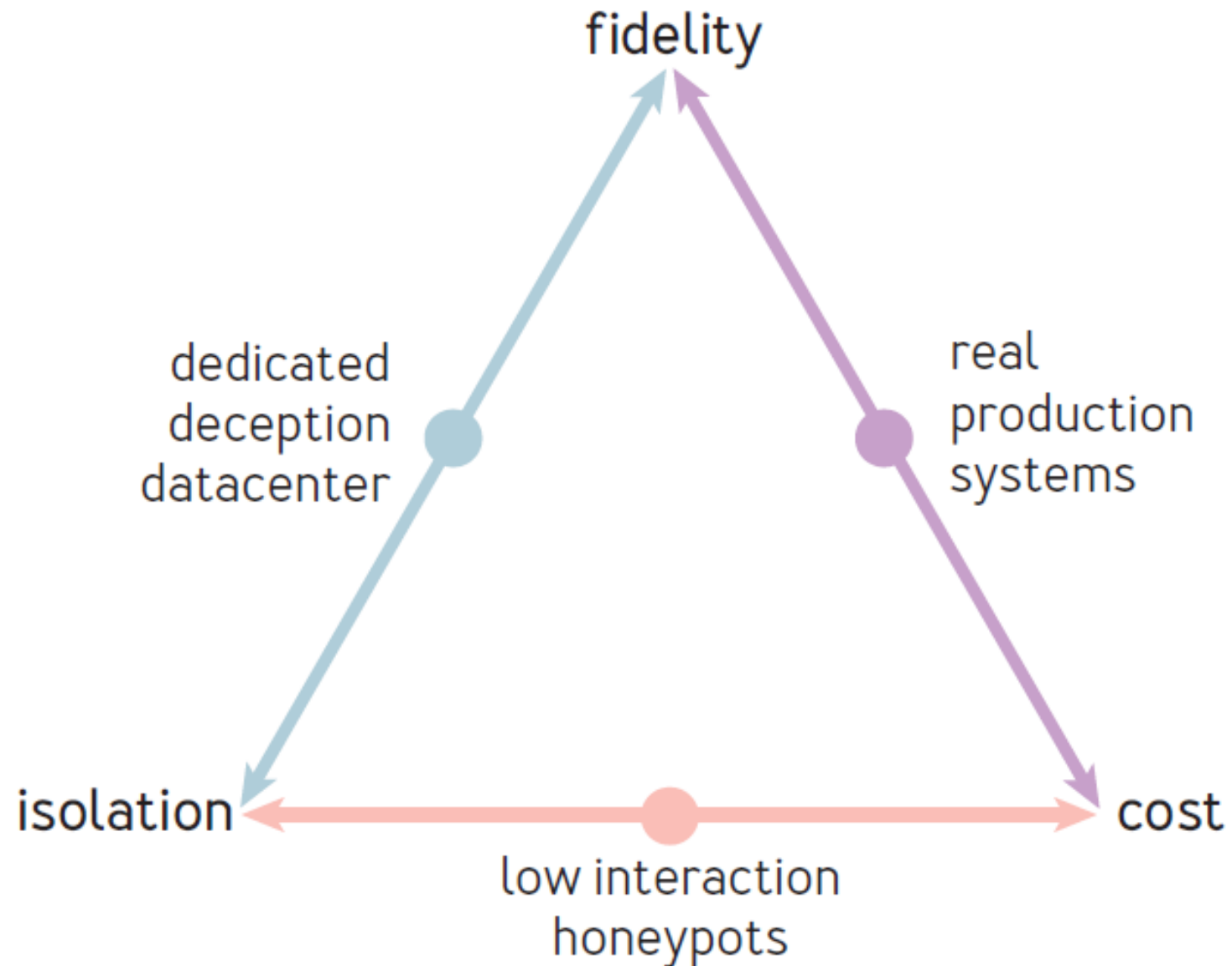
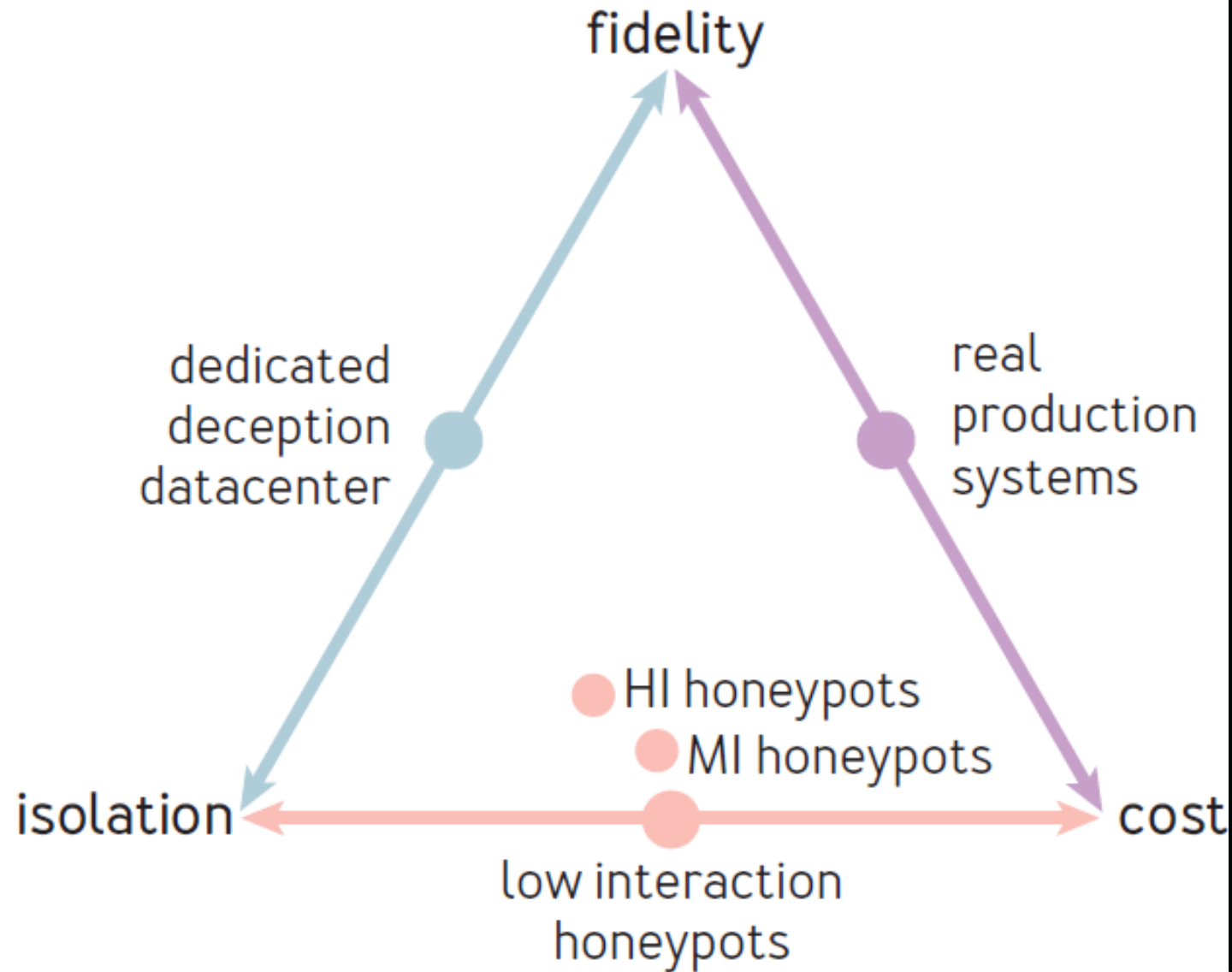
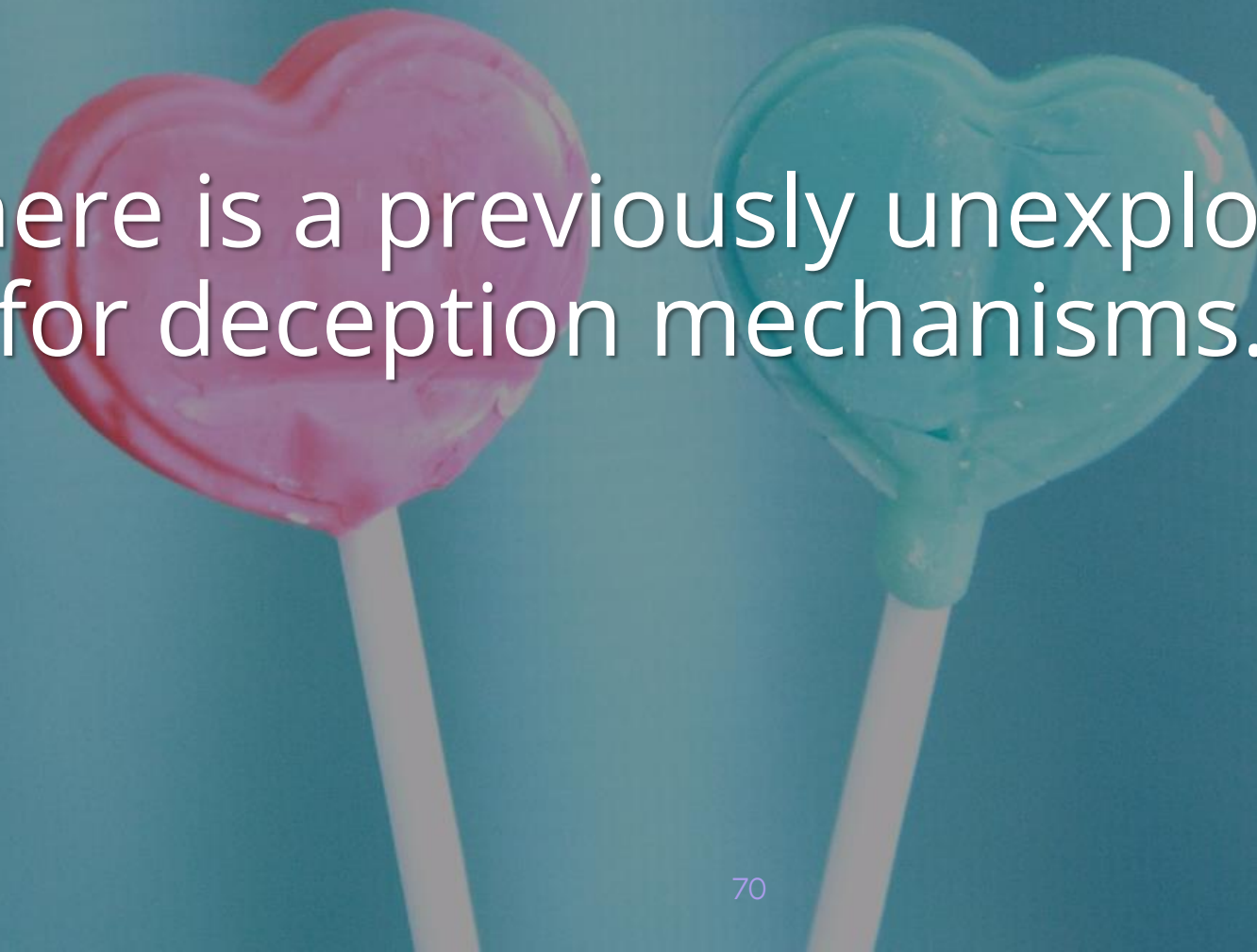


FIGURE 3: **MI & HI HONEYPOTS ON THE TRILEMMA**

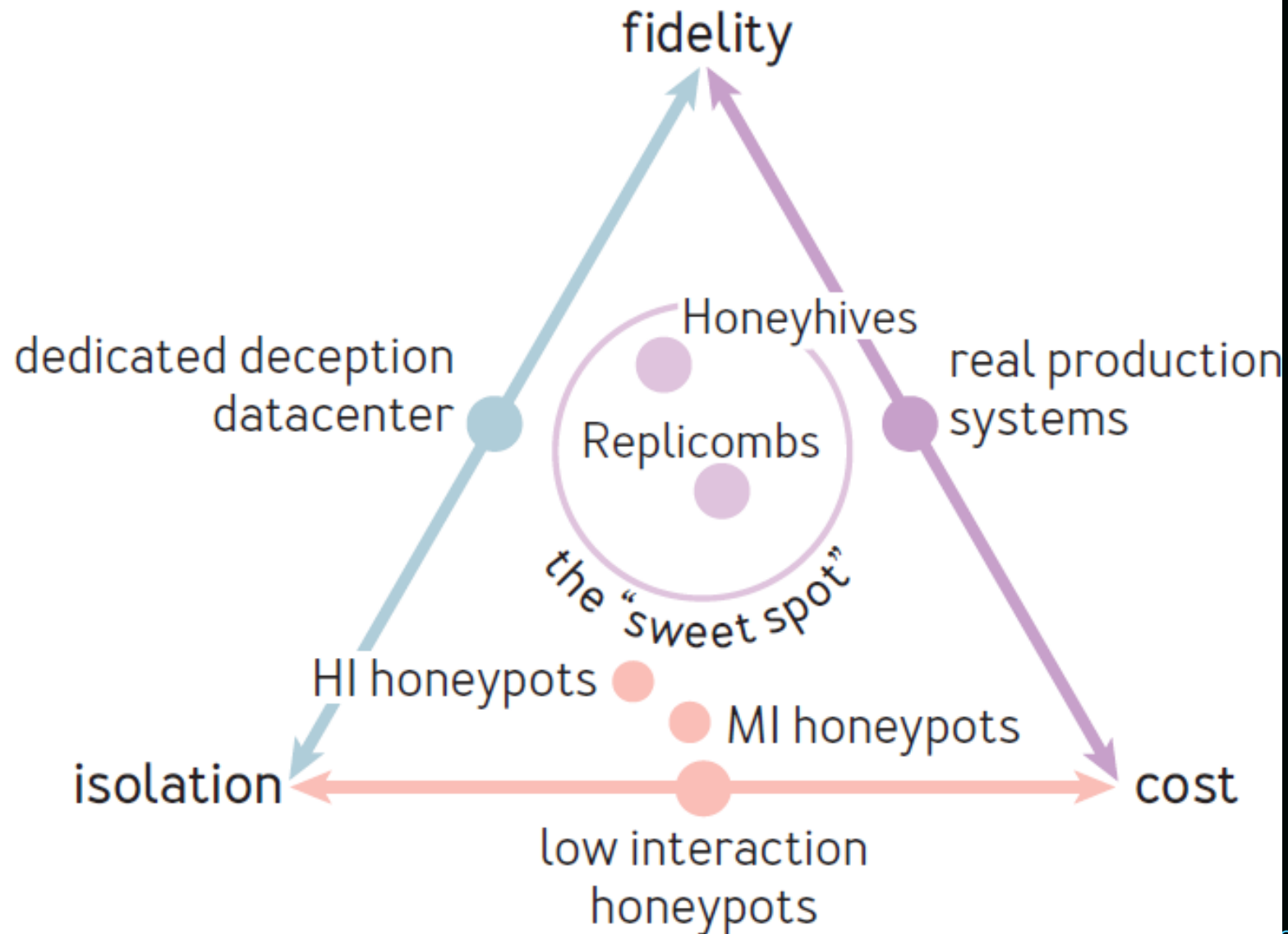


Two heart-shaped lollipops are positioned in the lower half of the frame. The one on the left is pink, and the one on the right is teal. They are set against a solid teal background. The text 'But there is a previously unexplored “sweet spot” for deception mechanisms...' is overlaid on the upper half of the image, centered horizontally and partially overlapping the lollipops.

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

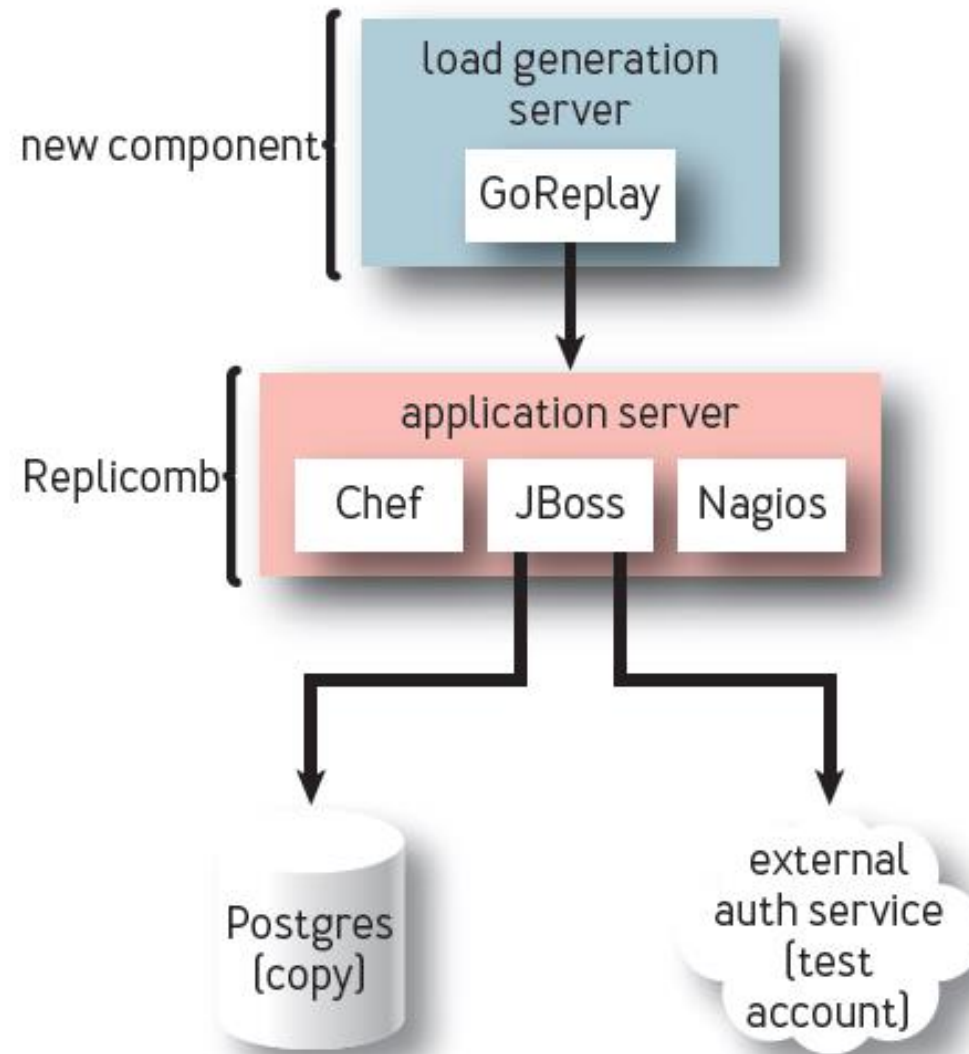


FIGURE 4: THE FIG "SWEET SPOT": HONEYHIVES AND REPLICOMBS



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

FIGURE 5: AN EXAMPLE REPLICOMB DEPLOYMENT

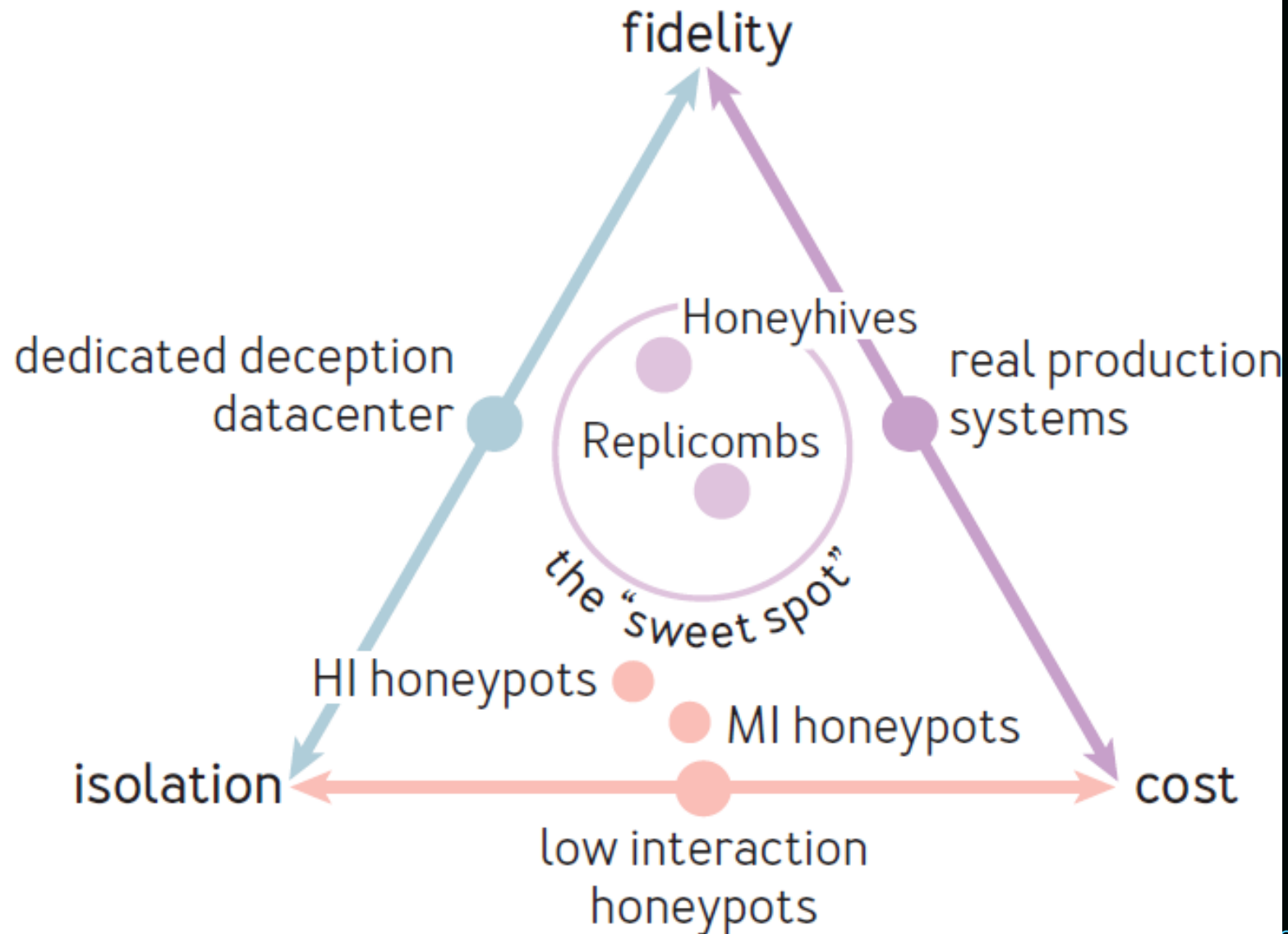






Replicomb vs. honeypot: impressive fidelity  
with an expansive range of attack behavior

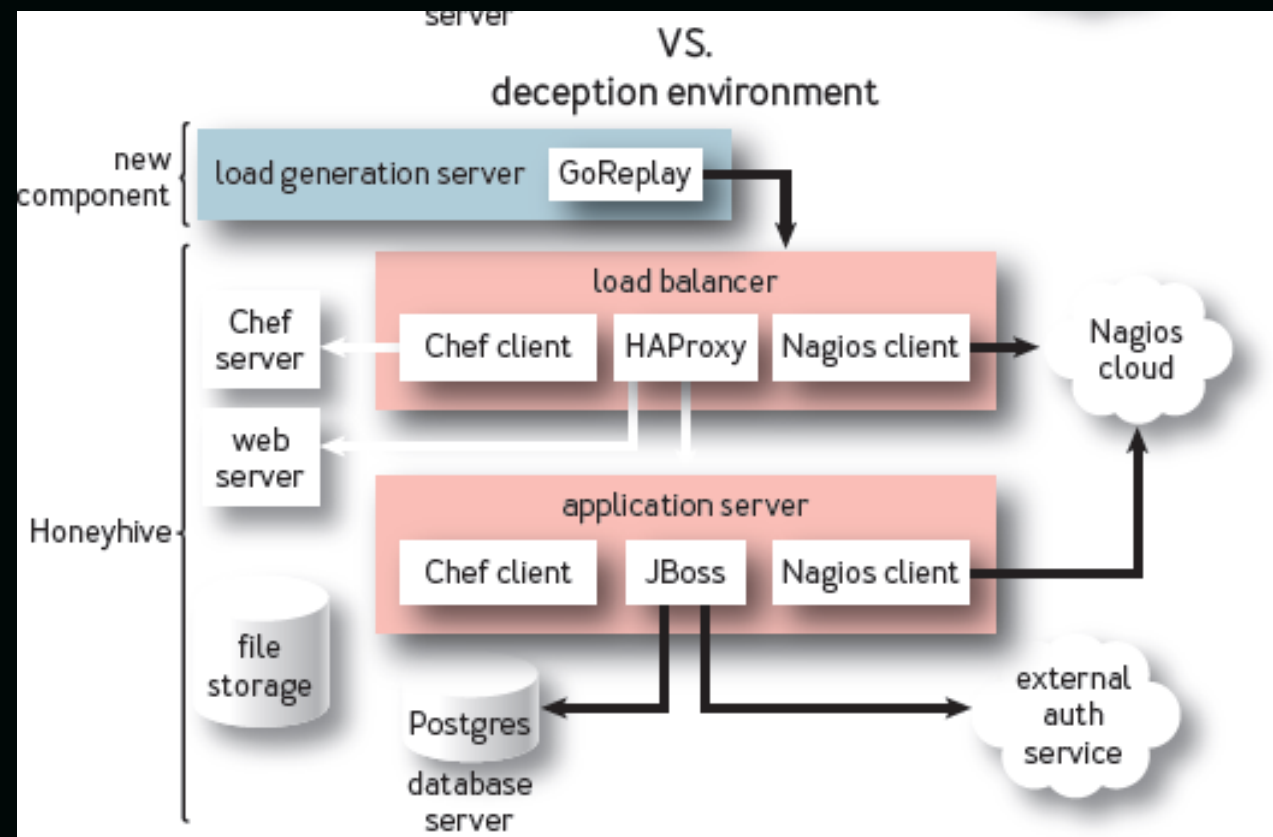
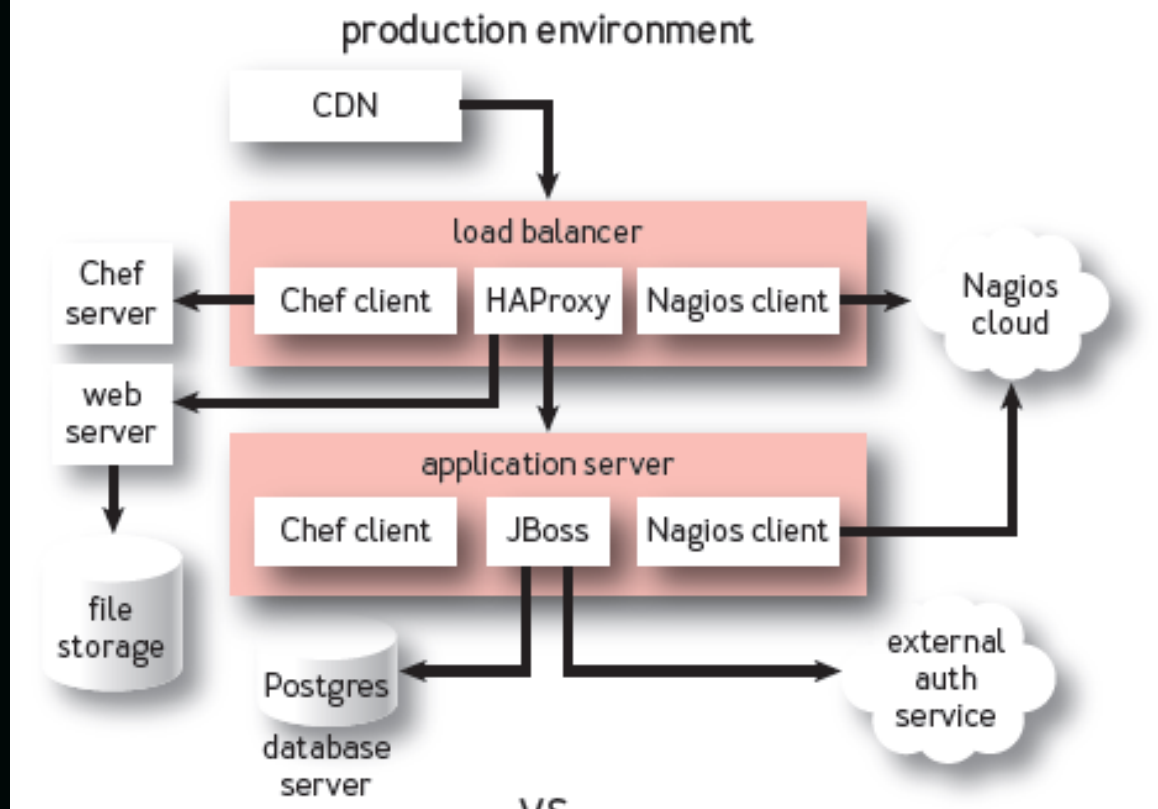
FIGURE 4: THE FIG "SWEET SPOT": HONEYHIVES AND REPLICOMBS



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

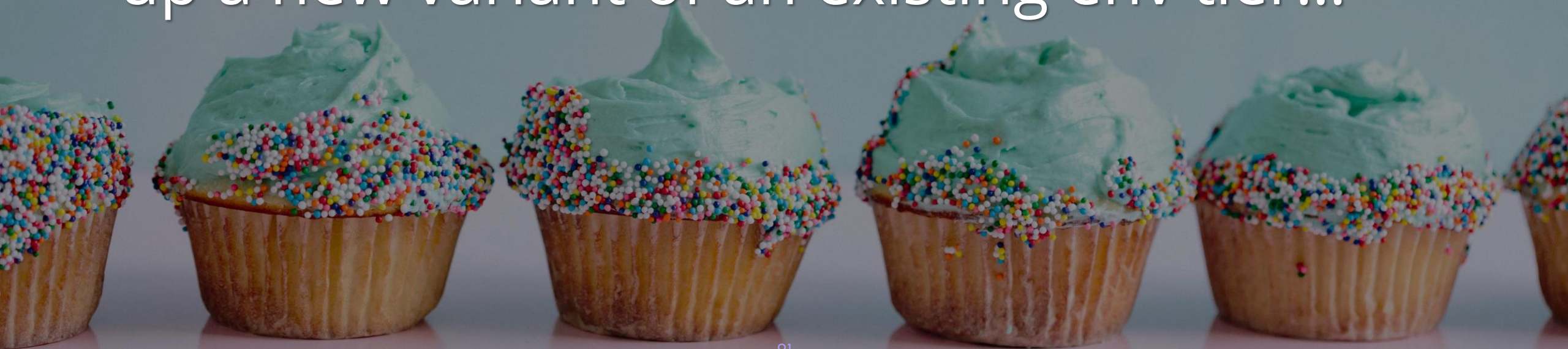


Honeyhives only need simulated load via a Replicomb as the initial entry point



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

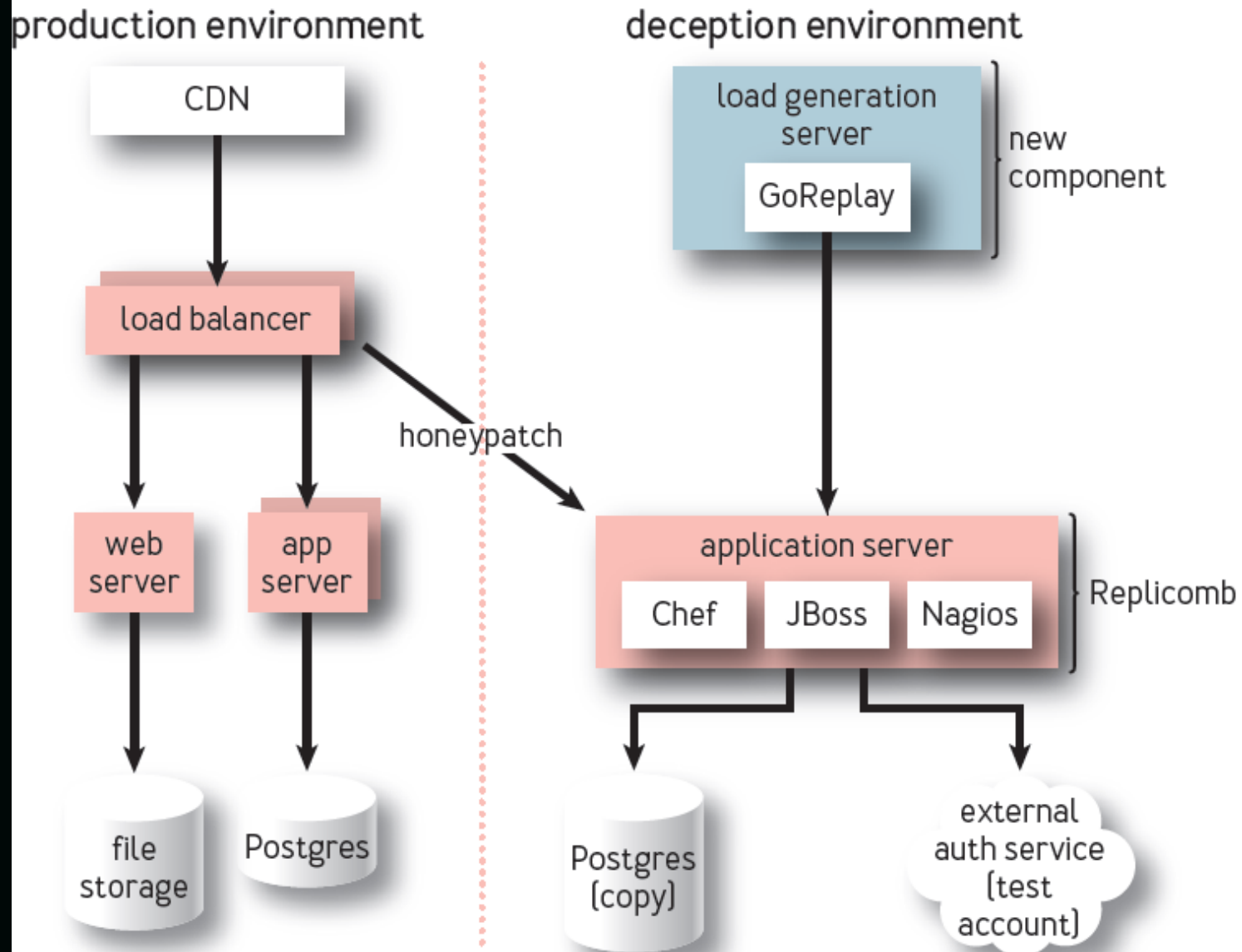




Discoverability: attackers need to find the DE for you to collect real data on their ops

# Honeypatching can support discoverability

FIGURE 7: AN EXAMPLE REPLICOMB ENVIRONMENT WITH HONEYPATCHING







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

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



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 hand wearing a pink nitrile glove holds a small, round, white object with pink horizontal stripes. The object has a dark circular 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

A vibrant rainbow arches across a blue sky with scattered white clouds. The rainbow's colors are bright and distinct, transitioning from purple on the left to red on the right. The sky is a deep blue, and the clouds are soft and white.

# 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



DEs expose opportunities for architectural improvements in operability & simplicity



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



# Attacker tracing

Attack observability enables pragmatic  
threat modeling during design & planning





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:

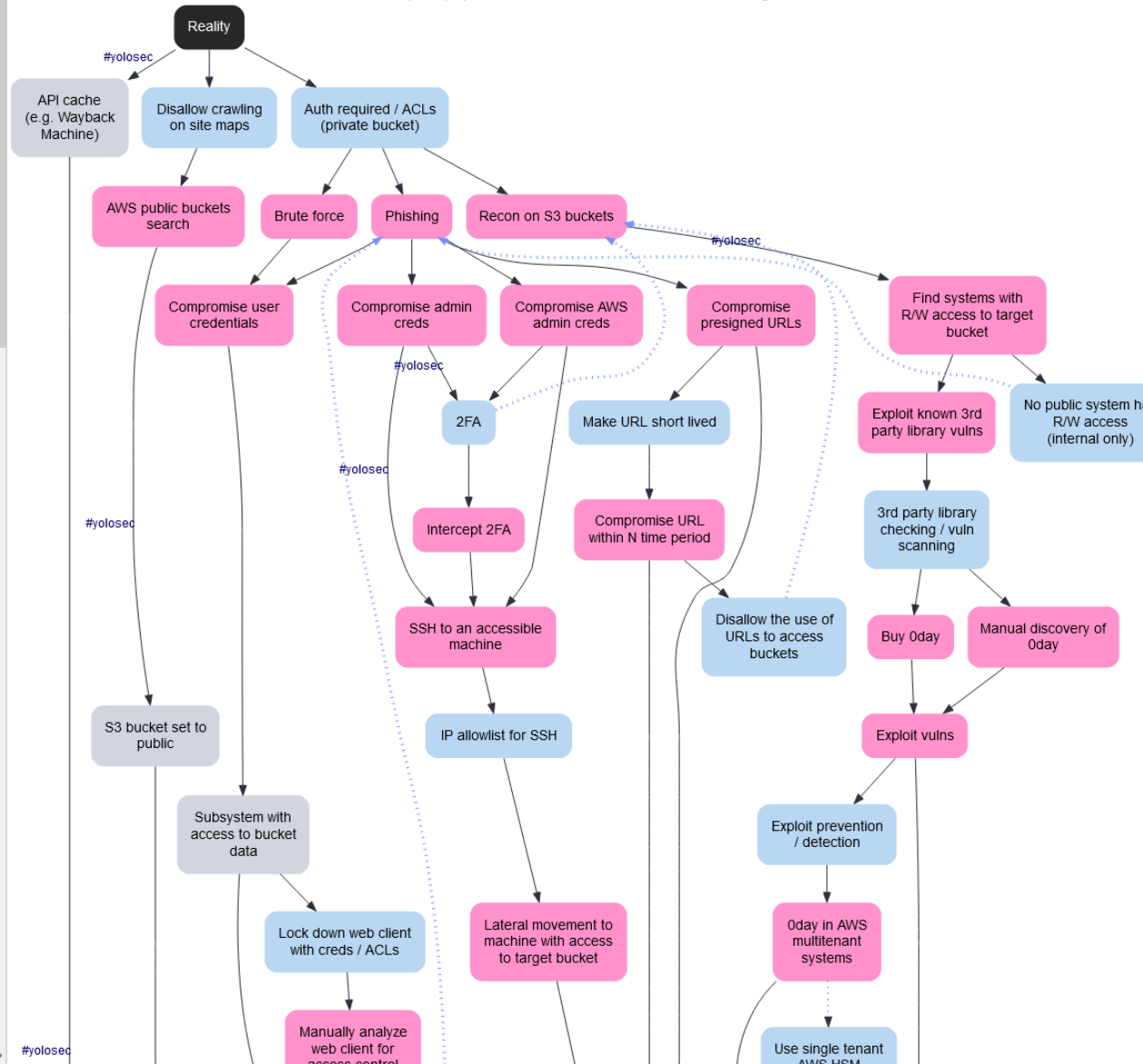
- **bucket\_search**: AWS public buckets search  
from:
  - **disallow\_crawling**
- **brute\_force**:  
from:
  - **private\_bucket**
- **phishing**:  
from:
  - **private\_bucket**
  - **internal\_only\_bucket**:  
backwards: true
  - **access\_control\_server\_side**:  
backwards: true
- **compromise\_user\_creds**: 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](#)

(Example) Attack Tree for S3 Bucket with Video Recordings







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



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





# VI. Future opportunities





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)



A photograph of the International Space Station (ISS) in orbit above Earth. The station's complex structure, including the large solar panel arrays and the Canadian module labeled 'Canada', is visible against the blackness of space. The Earth's surface below is a mix of green land and blue oceans, with a thin white atmospheric layer separating the planet from the station. The text 'Hypervisor-based observability' is overlaid in white on the lower half of the image.

# 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...





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 full-page background image featuring a powerful waterfall with turquoise water cascading over dark rocks. A bright, multi-colored rainbow is visible, arching across the upper half of the frame. The water is turbulent, with white foam and rapids. The overall scene is dramatic and natural.

# VI. Conclusion



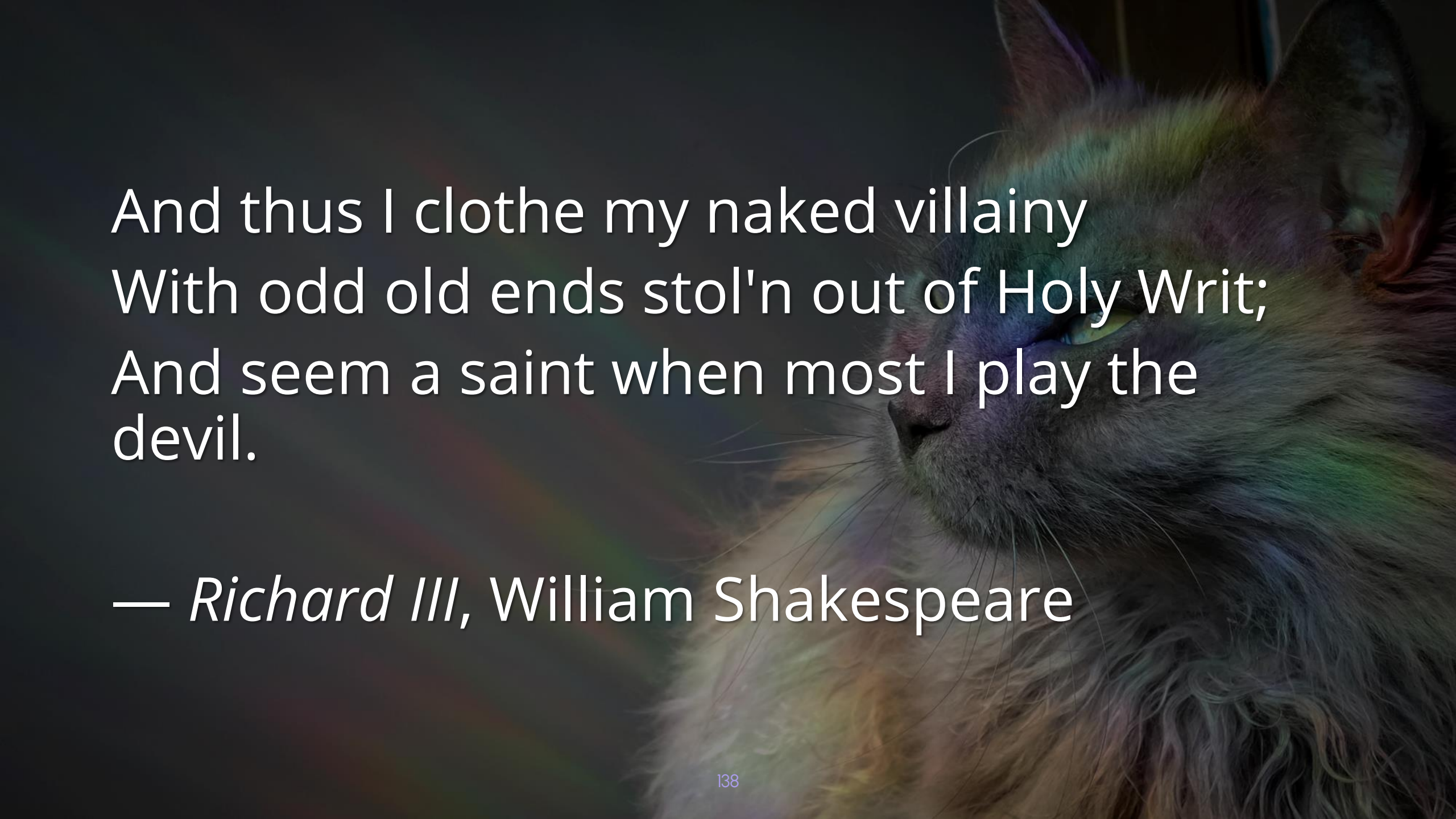
Imagine if SWEs could exploit attackers as much as attackers exploit defenders now!





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