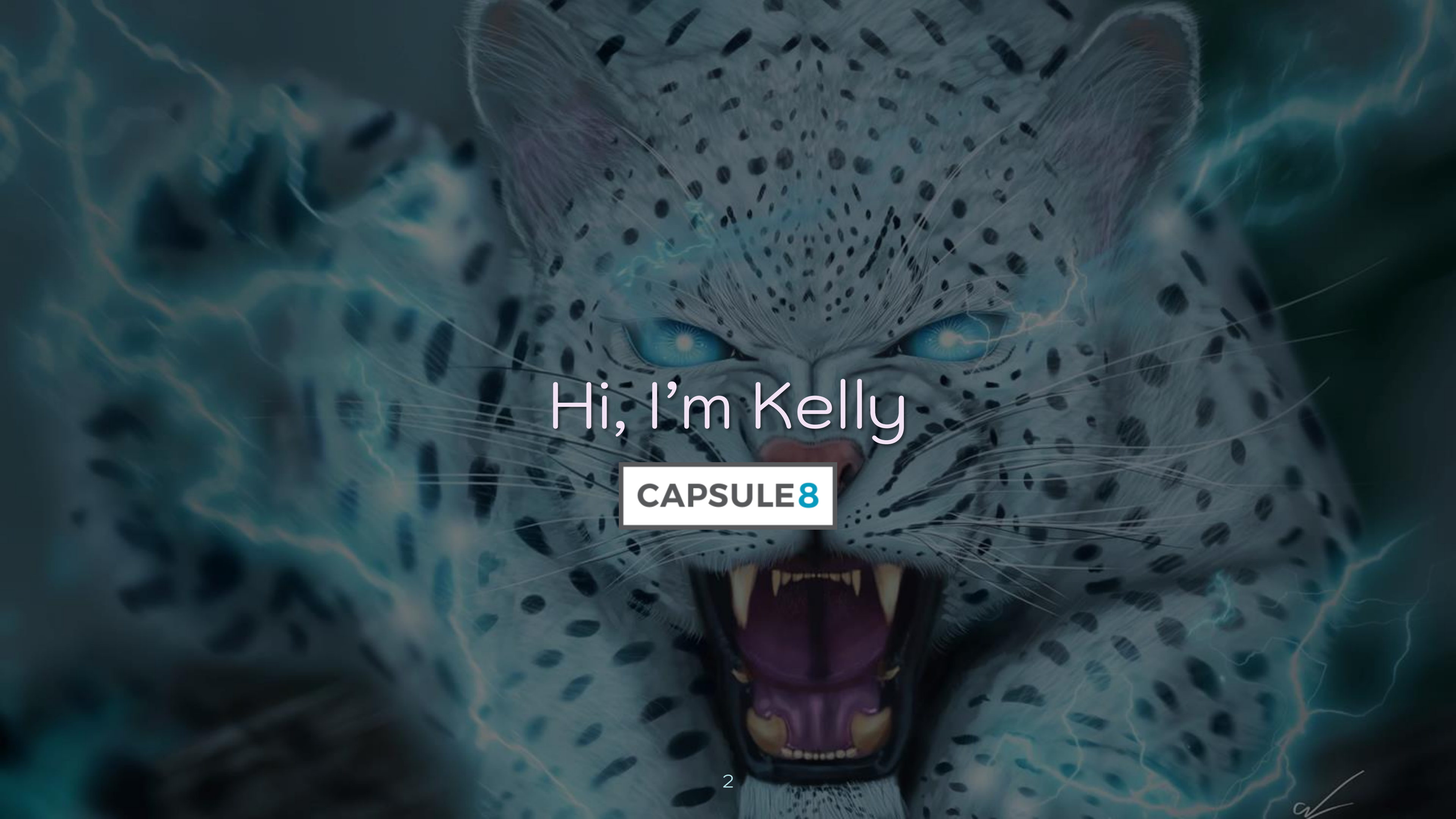
Controlled Chaos

The Inevitable Marriage of DevOps & Security

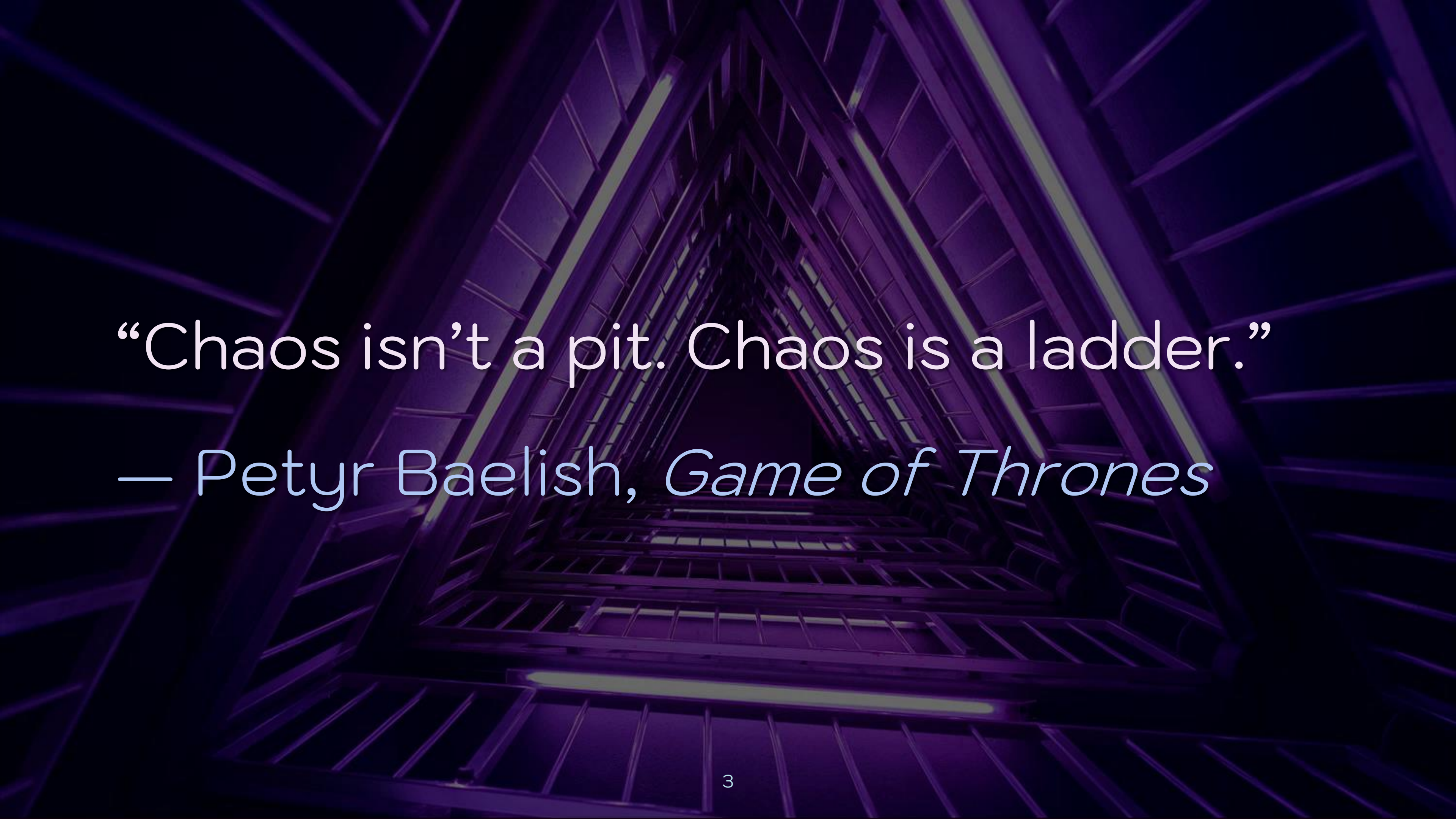
Kelly Shortridge (@swagitda_)

S4x20



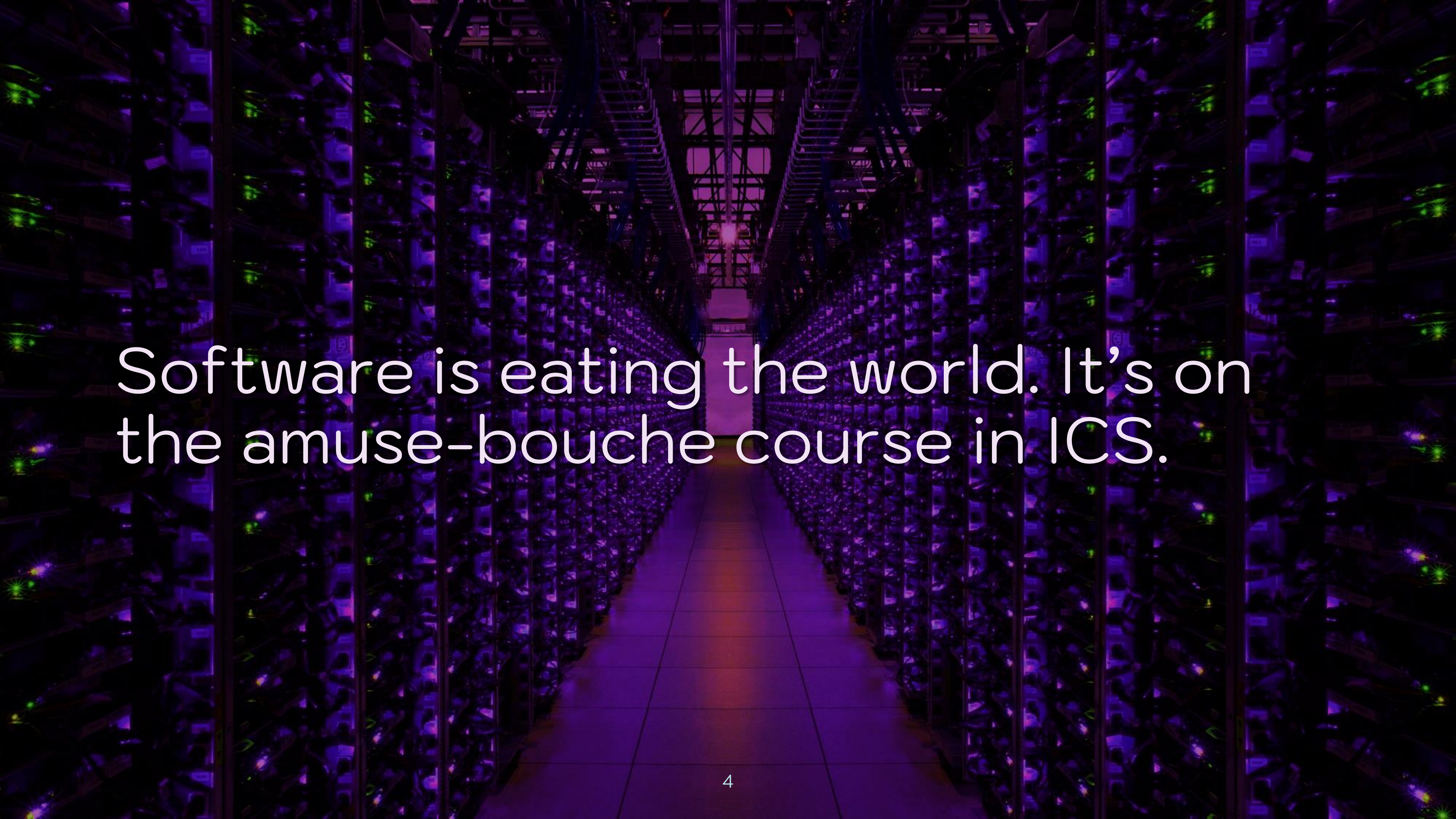
Hi, I’m Kelly

CAPSULE8



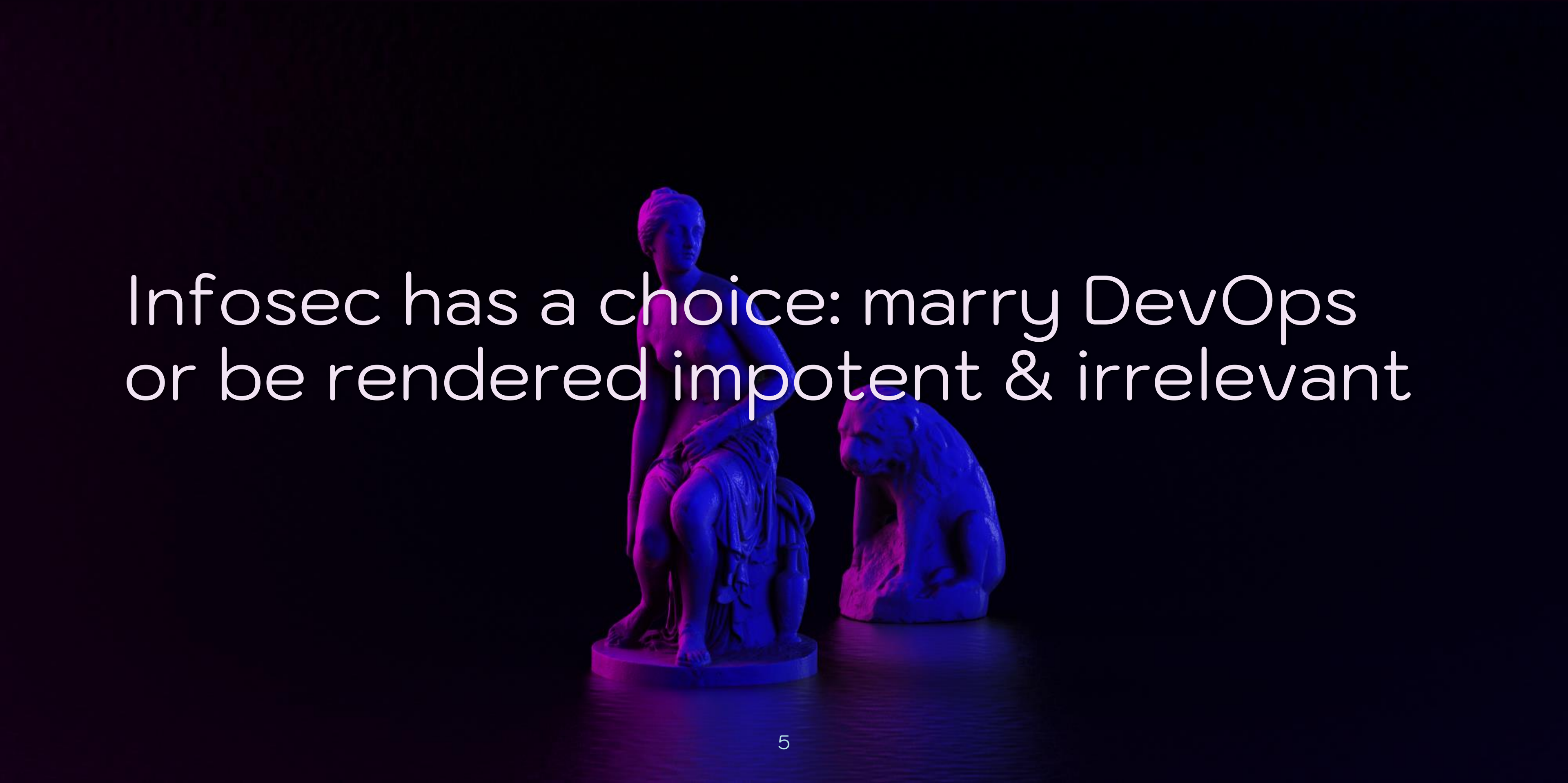
“Chaos isn’t a pit. Chaos is a ladder.”

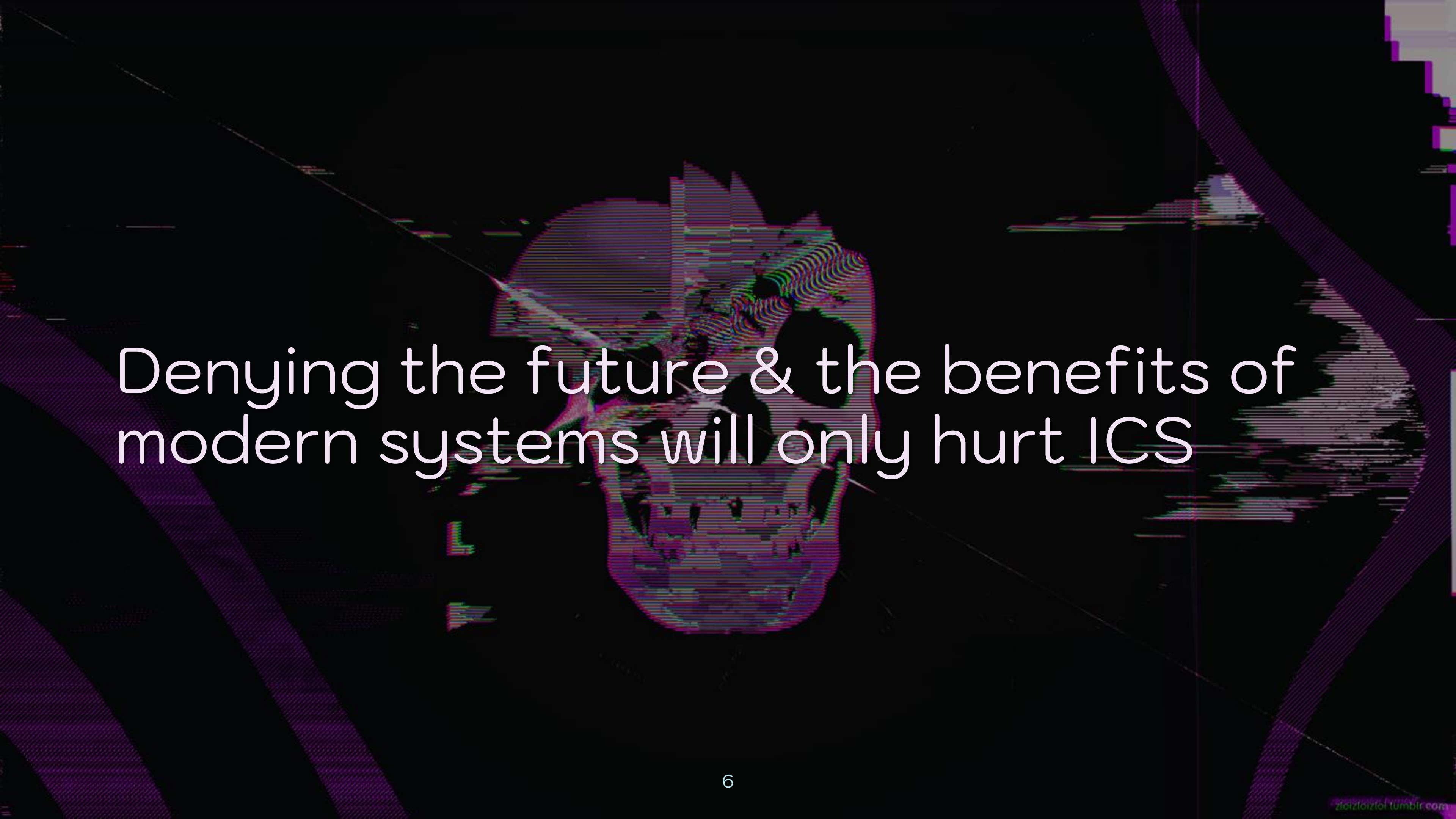
— Petyr Baelish, *Game of Thrones*



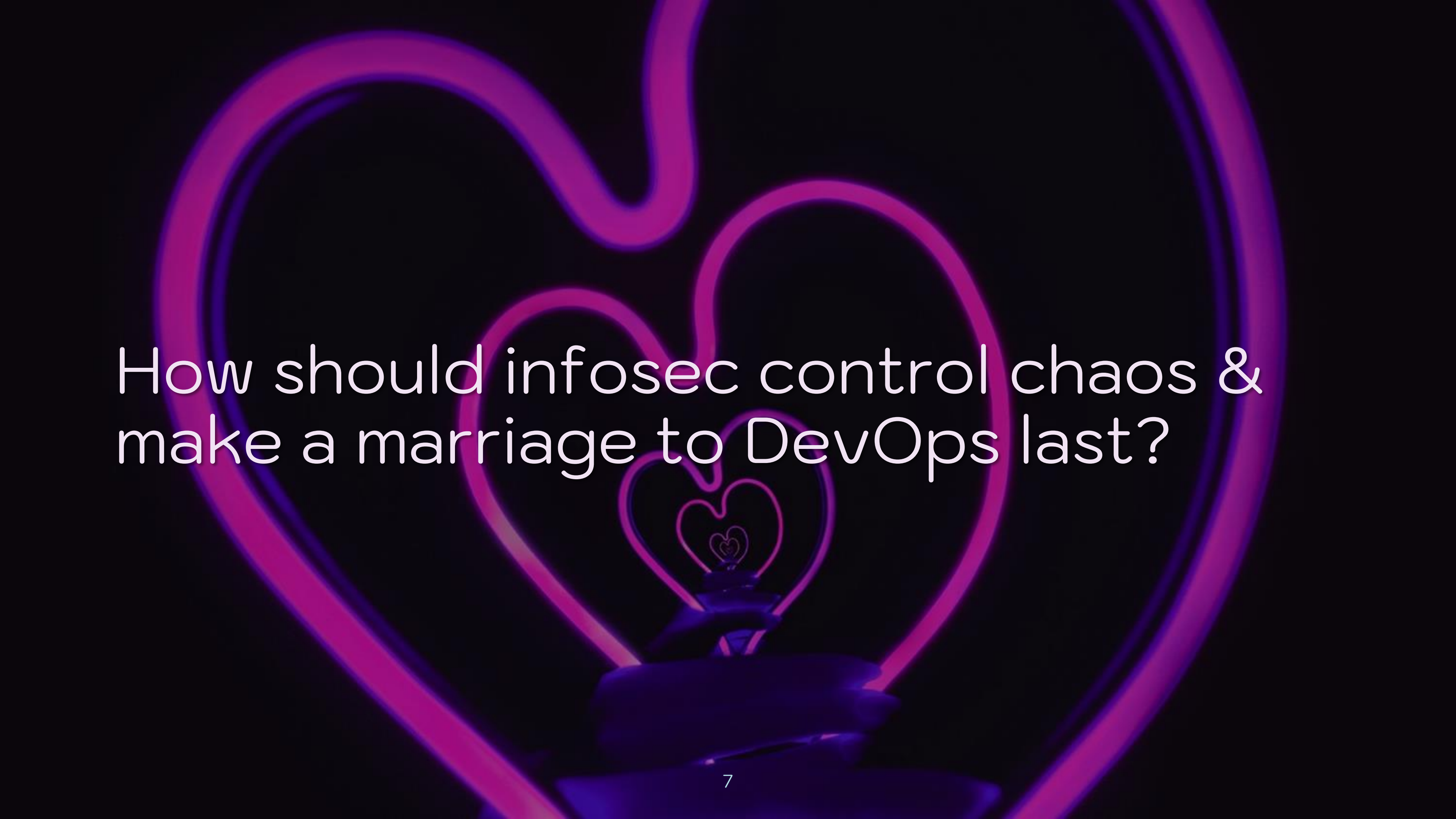
Software is eating the world. It’s on the amuse-bouche course in ICS.

Infosec has a choice: marry DevOps or be rendered impotent & irrelevant





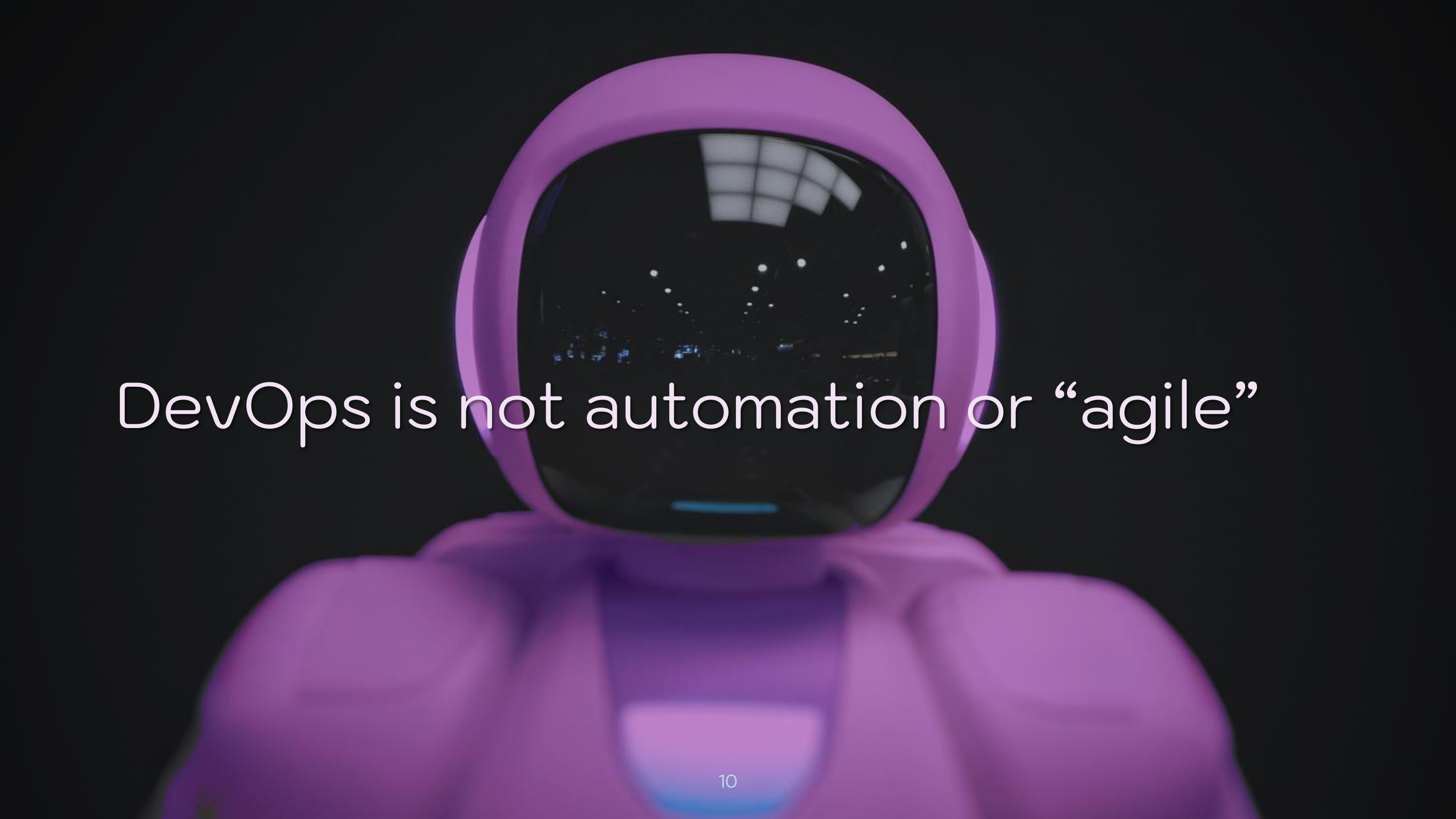
Denying the future & the benefits of modern systems will only hurt ICS



How should infosec control chaos & make a marriage to DevOps last?

1. DevOps Dominion
2. The Metamorphosis
3. Time to D.I.E.
4. A Phoenix Rises

DevOps Dominion



DevOps is not automation or “agile”

DevOps is a mindset that unifies responsibility and accountability.

Infosec can join DevOps or take a back seat to the future of systems

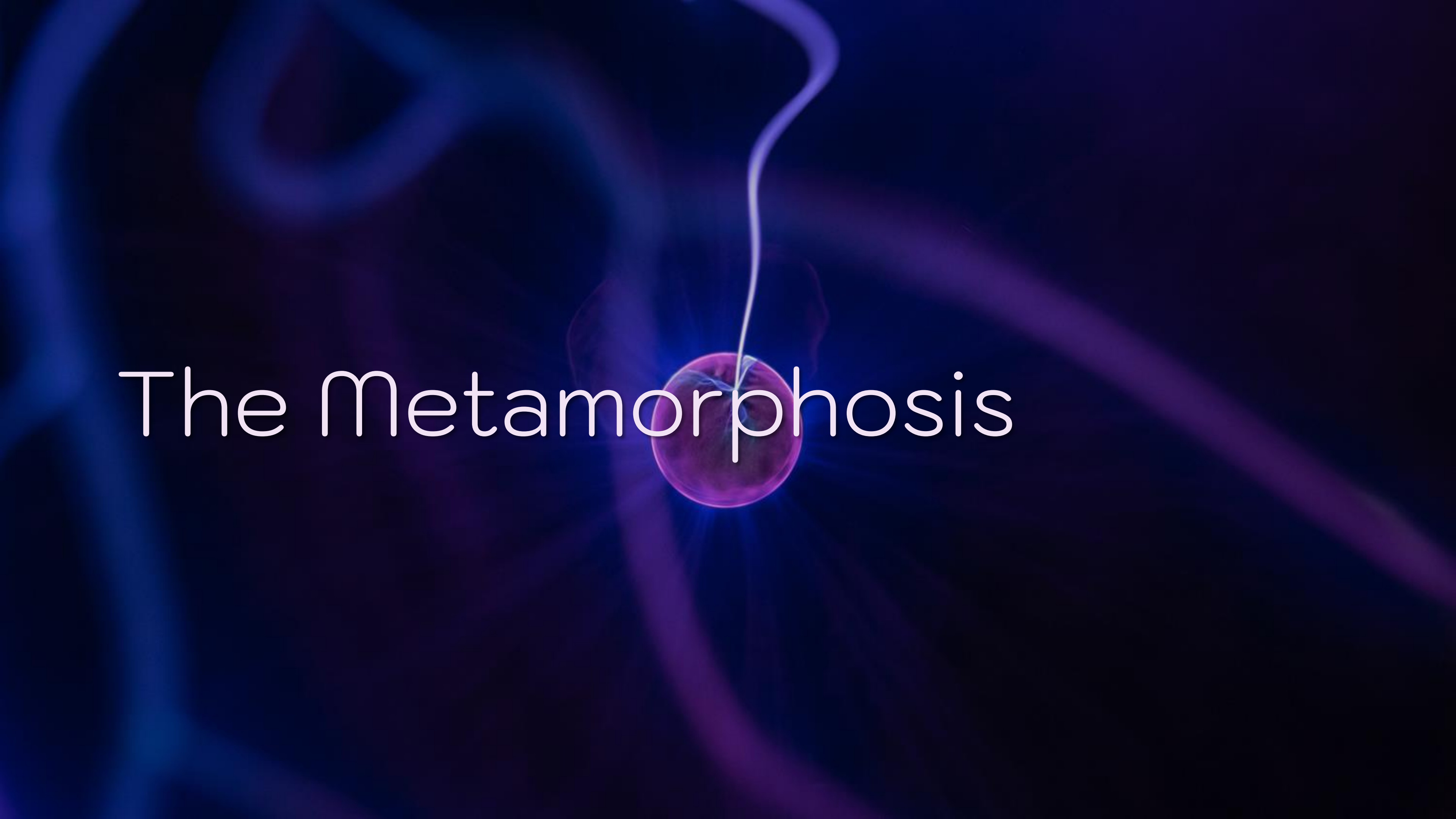
Chaos & resilience is infosec’s future

What are DevOps’s priorities?

Optimization of software delivery performance so tech delivers value

Stability & speed don’t conflict – resilience & innovation are bffs

Security drives stronger DevOps results. Now ICS security must evolve.



The Metamorphosis

Partitioning of responsibility & accountability engenders conflict

After this evolution, DevOps will be held accountable for security fixes

What goals should infosec pursue in this evolution?

And… why should infosec goals diverge from DevOps goals?

Infosec has arguably failed, so “this is how we’ve always done it” is invalid

The Security of Chaos

“Things will fail” naturally extends into “things will be pwned”

Security failure is when security controls don’t operate as intended

What are the principles of chaotic security engineering?

1. Expect that security controls will fail & prepare accordingly

2. Don’t try to avoid incidents – hone your ability to respond to them

What are the benefits of the chaos / resilience approach?

Benefits: lowers remediation costs & stress levels during real incidents

Benefits: minimizes service disruption & improves confidence

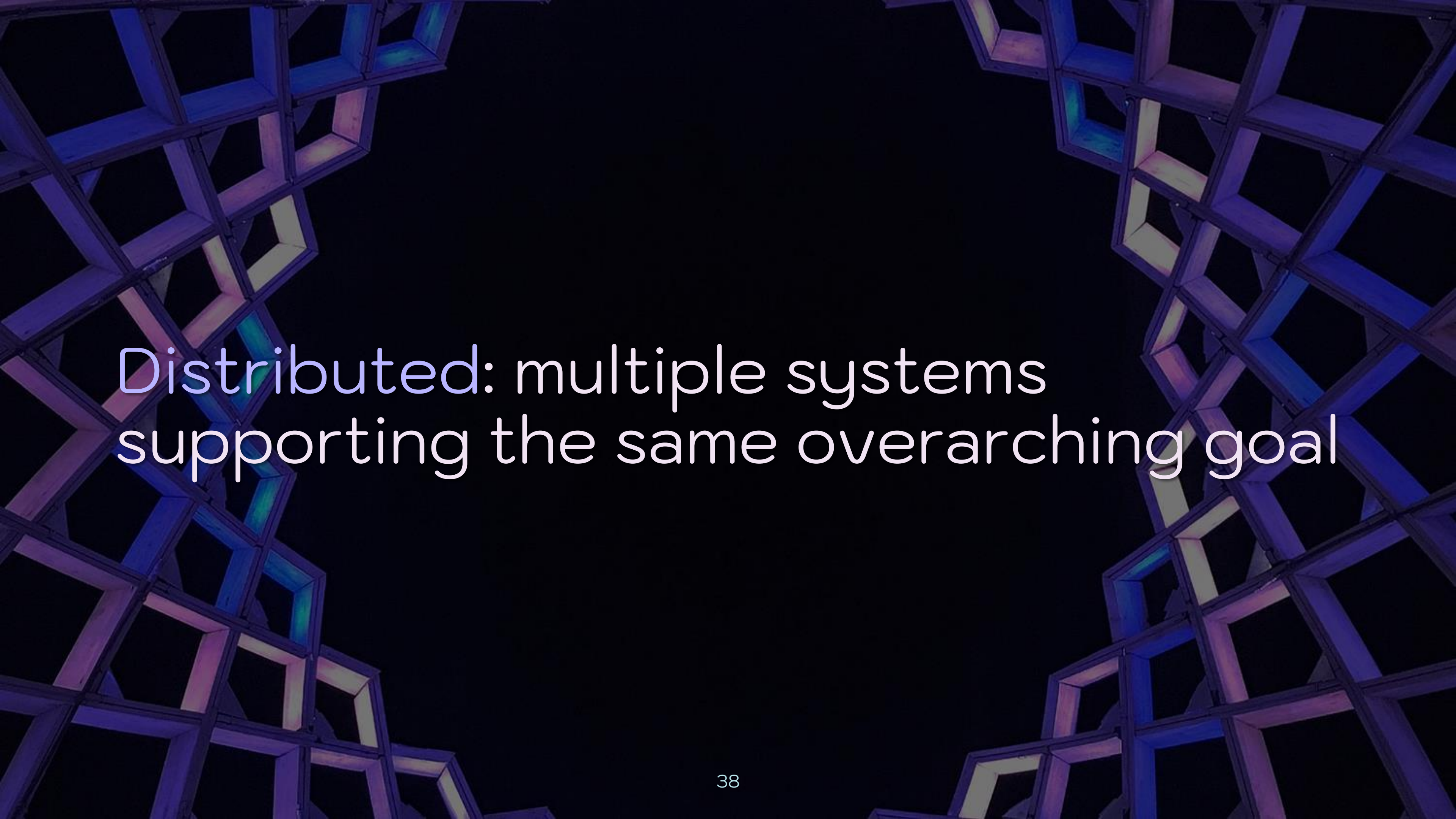
Benefits: creates feedback loops to foster understanding of systemic risk

What other ways can infosec become more strategic?

Time to D.I.E.

We need a model promoting qualities that make systems more secure

Enter the D.I.E. model: Distributed, Immutable, Ephemeral



Distributed: multiple systems supporting the same overarching goal

Distributed infrastructure reduces risk of DoS attacks by design

Immutable: infrastructure that doesn’t change after it’s deployed

Servers are now disposable “cattle” rather than cherished “pets”

Immutable infra is more secure by design – ban shell access entirely

Unlimited lives is better for security than game over upon death

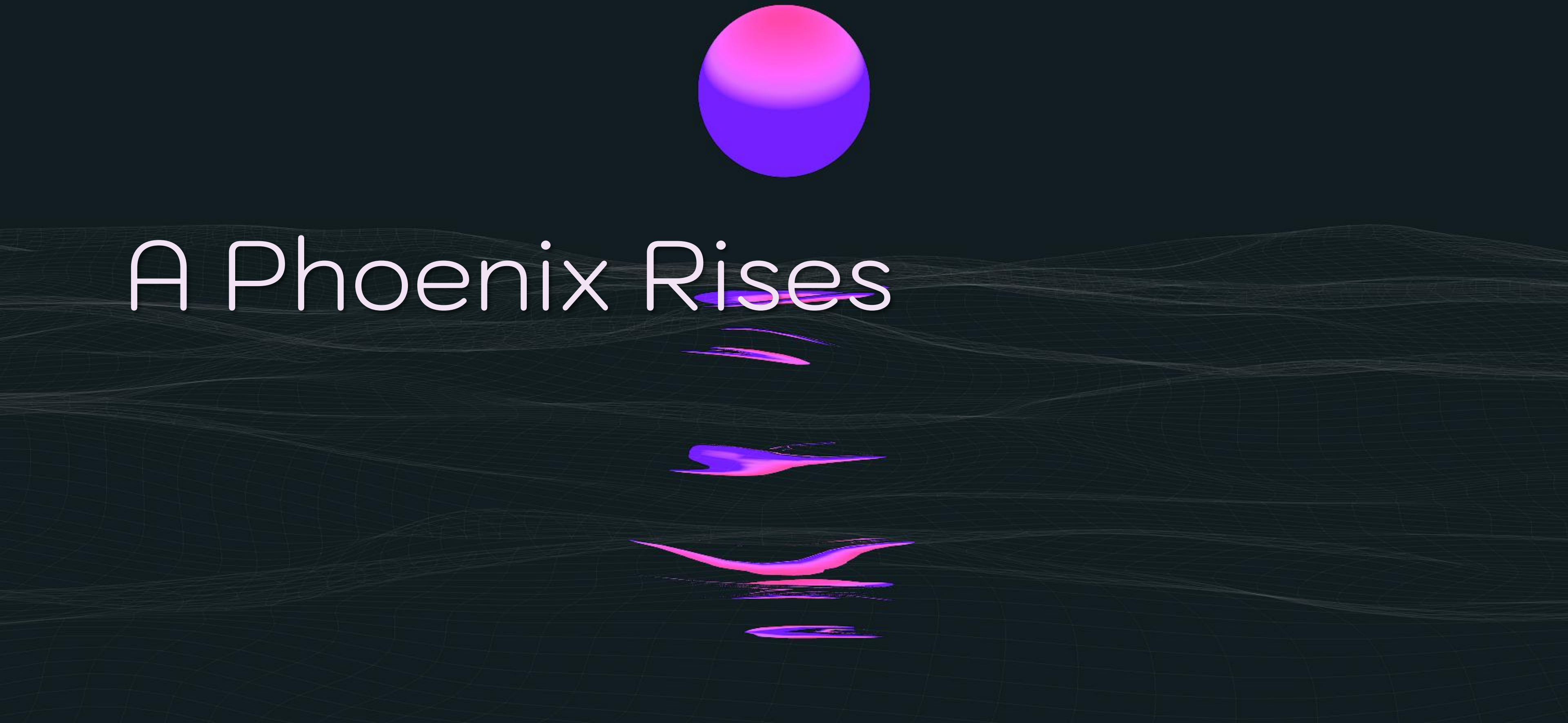
Ephemeral: infrastructure with a very short lifespan (dies after a task)

Ephemerality creates uncertainty for attackers (persistence = nightmare)

Installing a rootkit on a resource that dies in minutes is a waste of effort

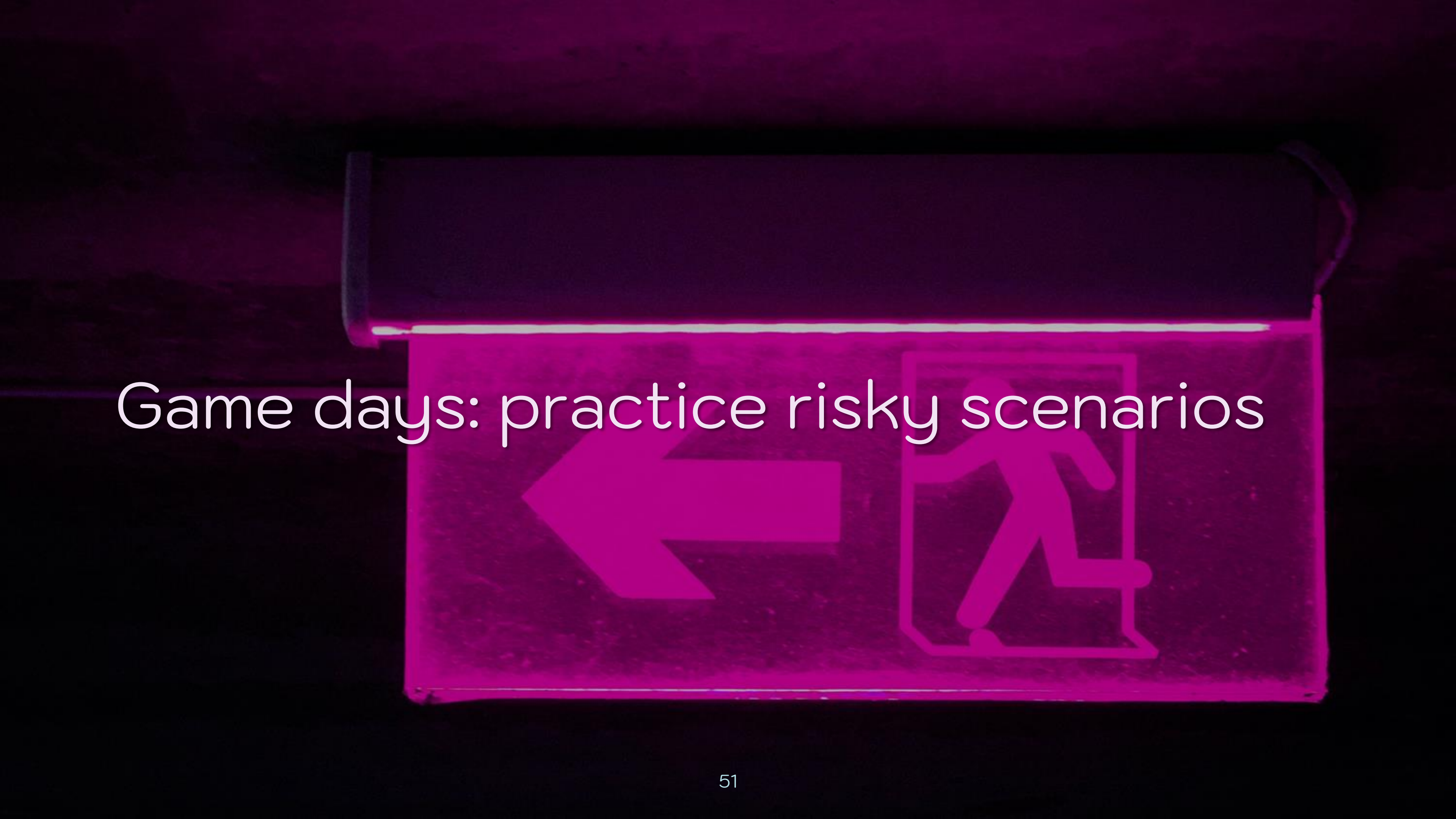
ICS attacks take months to plan; ephemerality constantly disrupts it

Optimizing for D.I.E. reduces risk by design & supports resilience



A Phoenix Rises

Harness failure as a tool to help you prepare for the inevitable



Game days: practice risky scenarios

Prioritize game days based on potential business impacts

Decision trees: start at target asset, work back to easiest attacker paths

Determine the attacker’s least-cost path (hint: it doesn’t involve 0day)

Architecting chaos

Begin with “dumb” testing before moving to “fancy” testing

Think digital twins, analytics services, or O365… *not* field-level SCADA

Controlling Chaos: Distributed

Distributed mostly overlaps with availability in modern infra contexts

Chaos Monkey: inject random instances failures to test resilience

Infosec teams can use these tools but make attackers the source of failure

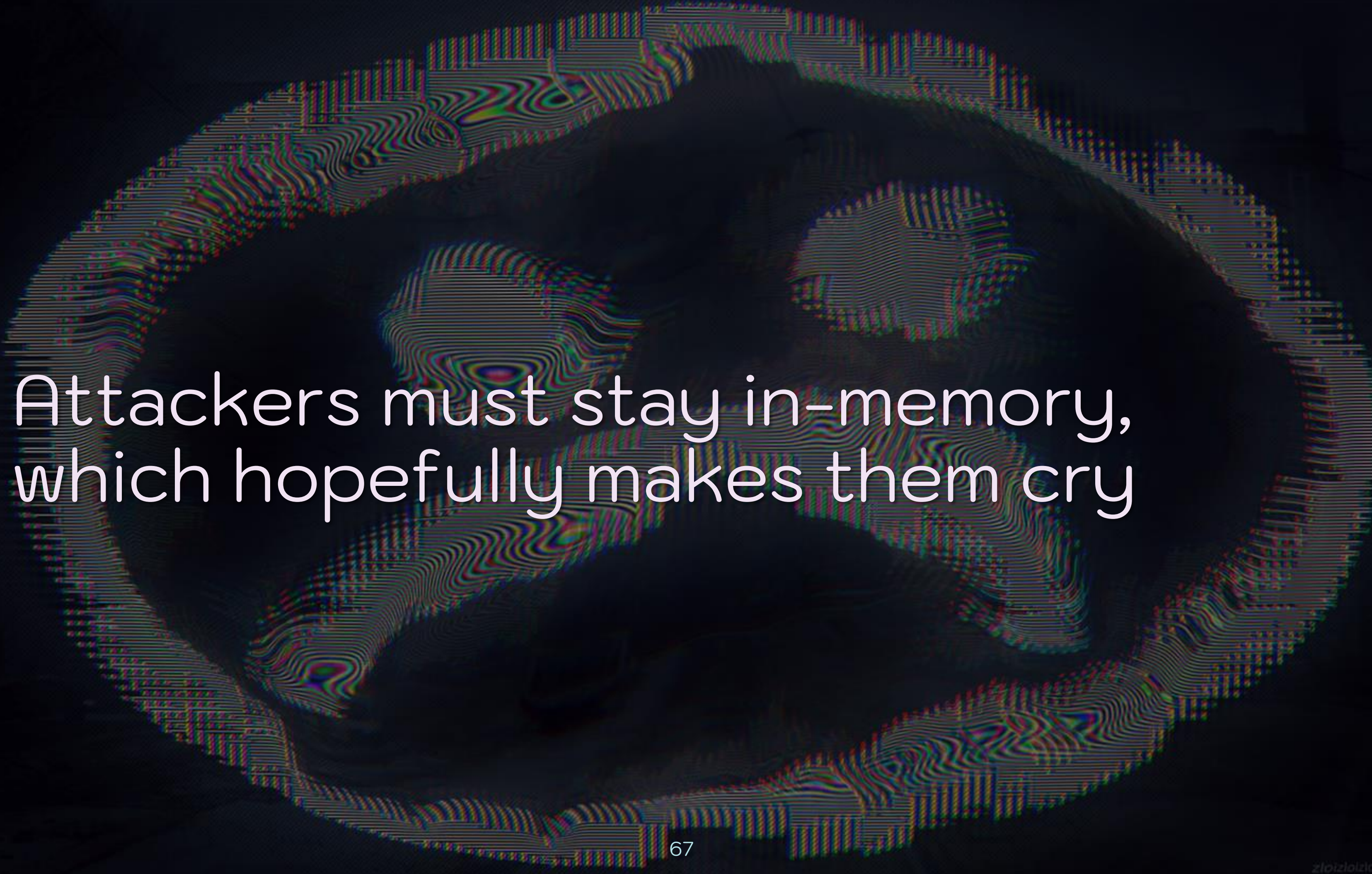
Multi-region services present a fun opportunity to mess with attackers

Shuffle IP blocks regularly to change attackers' lateral movement game

Controlling Chaos: Immutable

Volatile environments with continually moving parts raise the cost of attack

Create rules like, “If there’s ever a write to disk, crash the node”



Attackers must stay in-memory, which hopefully makes them cry

Metasploit Meterpreter + webshell:
Touch passwords.txt & kaboom

Infosec teams can build Docker images with a “bamboozle layer”

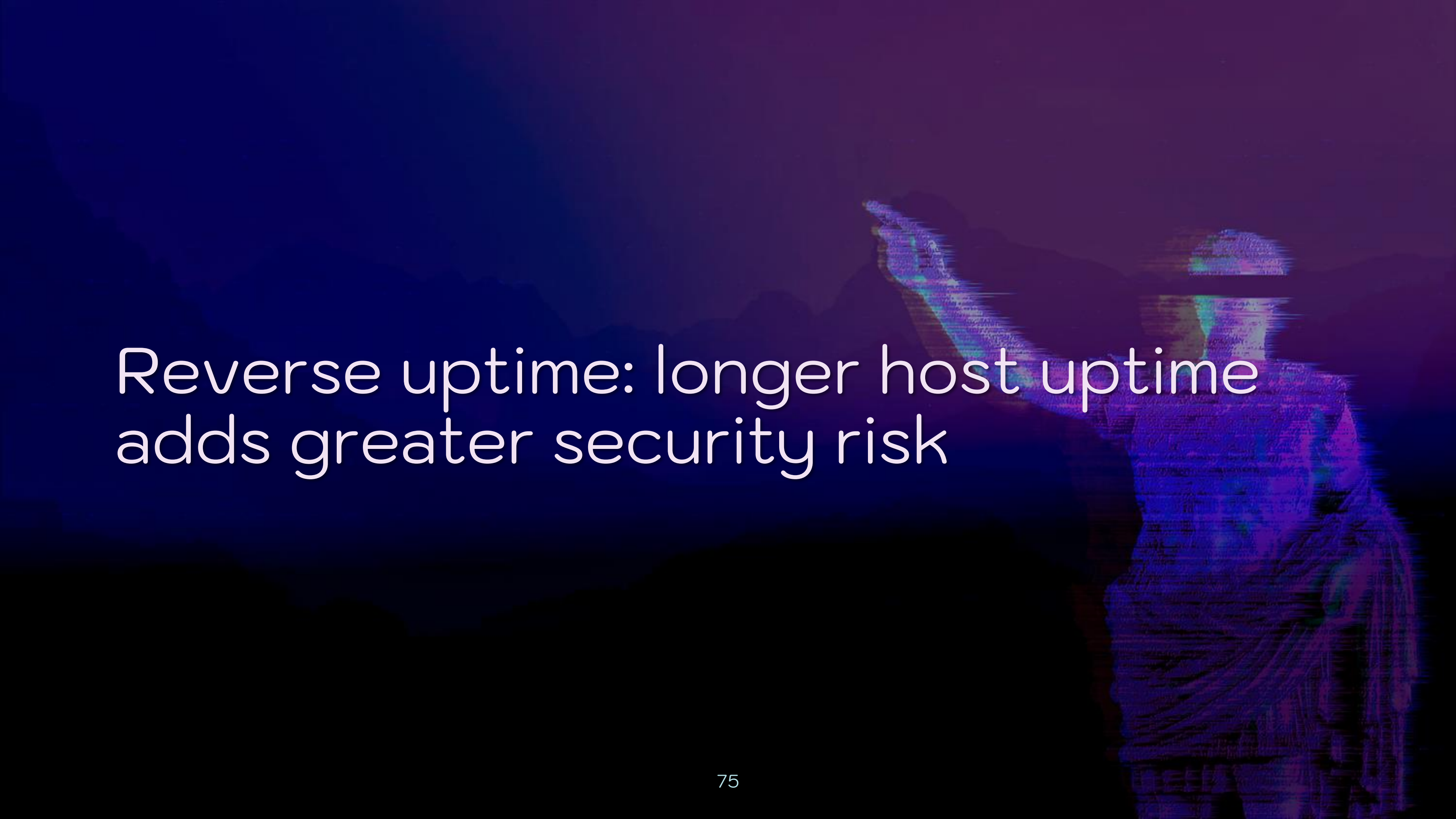
Mark garbage files as “unreadable” to craft enticing bait for attackers

Potential goal: self-healing edge devices with immediate reversion

Test: inject attempts at writing to disk to ensure detection & reversion

Controlling Chaos: Ephemeral

Most infosec bugs are stated-related – get rid of state, get rid of bugs



Reverse uptime: longer host uptime adds greater security risk

Test: retrograde libraries, containers, other resources in CI/CD pipelines

Leverage lessons from toll fraud – cloud billing becomes security signal

Test: exfil TBs or run a cryptominer to inform billing spike detection

Conclusion

Security cannot gatekeep DevOps.
It must marry it.

Chaos/resilience are natural homes for infosec & represent its future.

Infosec must now evolve to unify responsibility & accountability.

ICS is already cloudy – get ready now before OT migrates as well.

Giving up control isn’t a harbinger of doom. Resilience is a beacon of hope.

> “You must have chaos within you to give birth to a dancing star.”
>
> — Friedrich Nietzsche

@swagitda_

/in/kellyshortridge

kelly@greywire.net