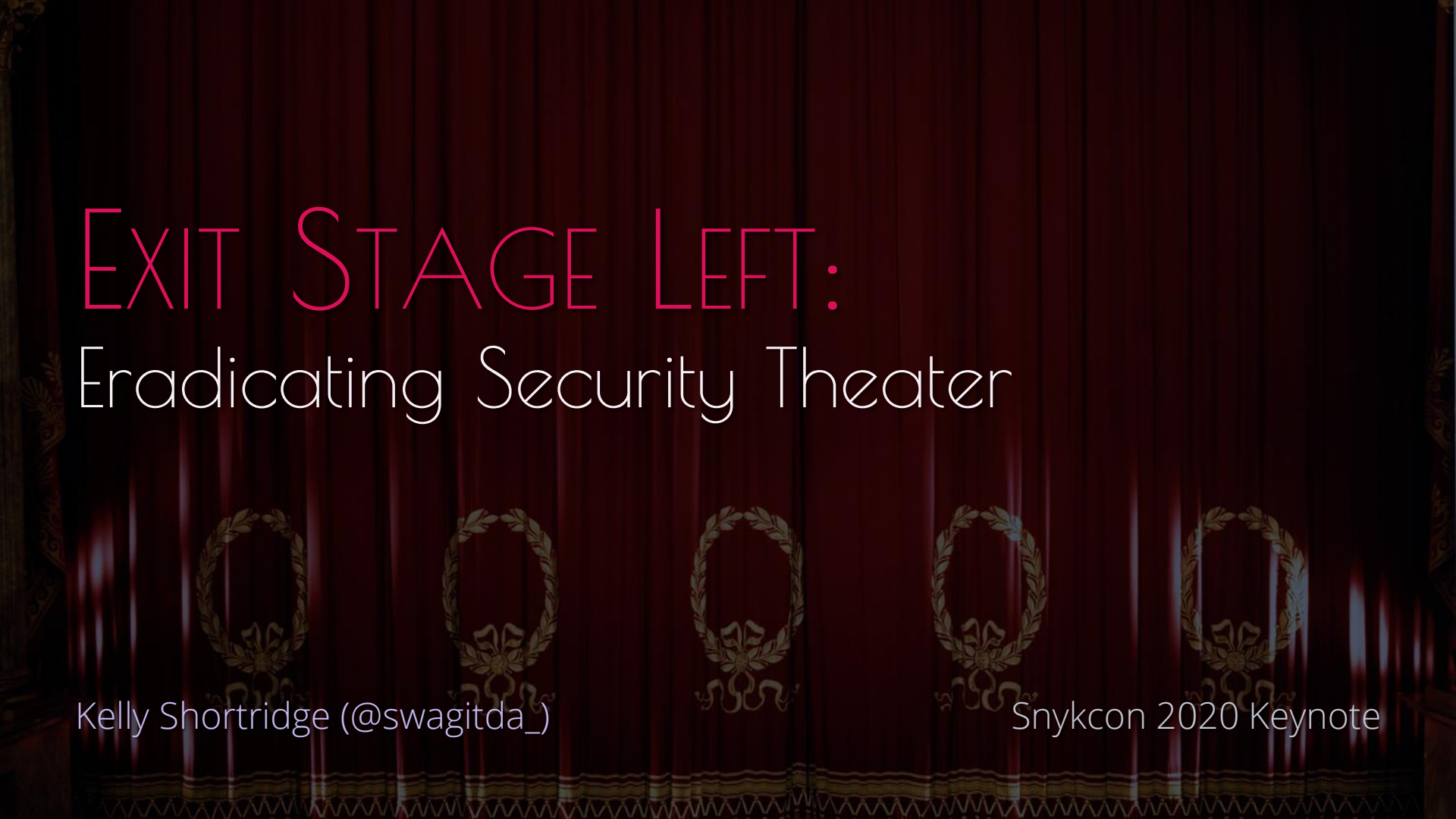
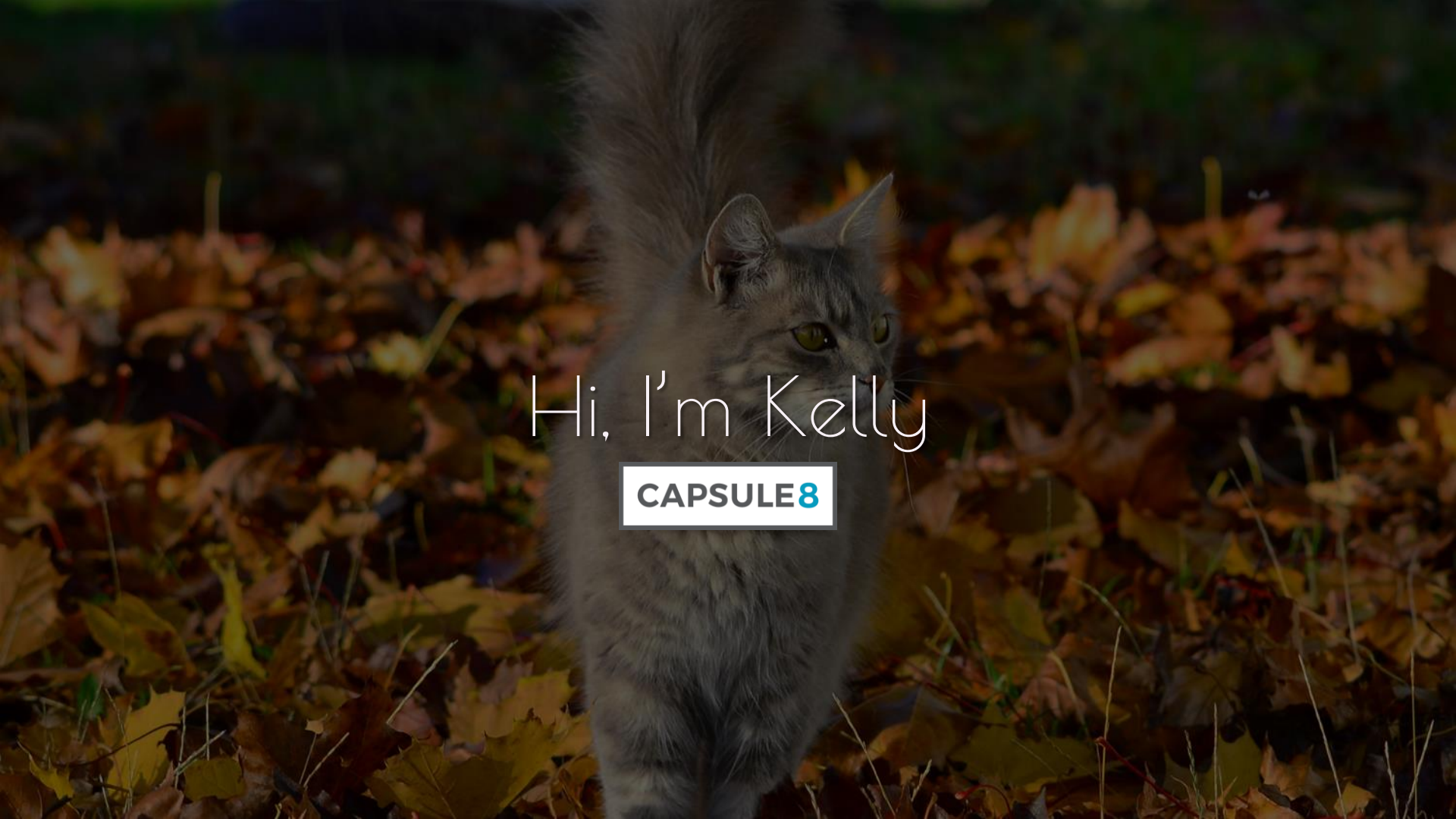
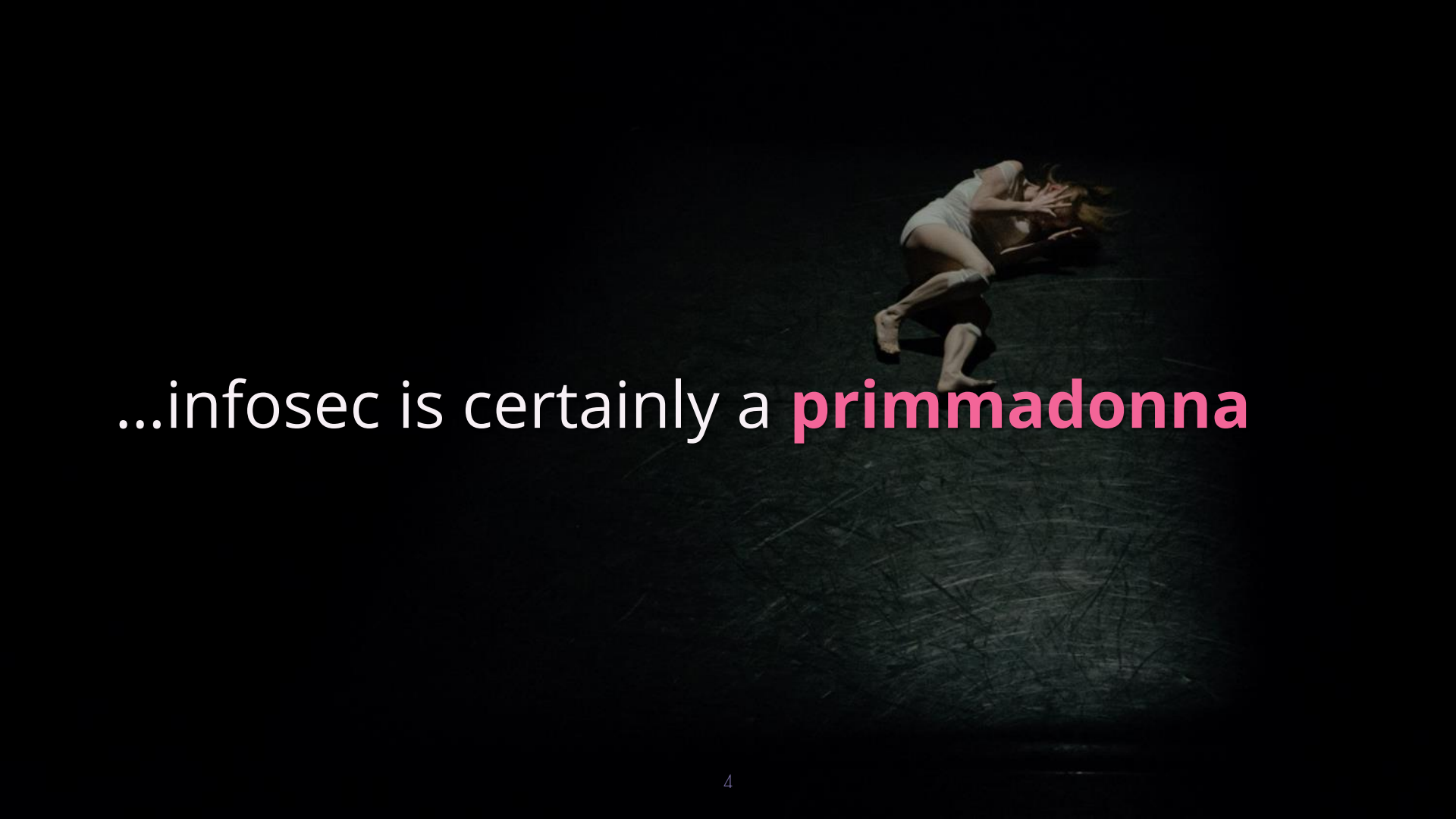
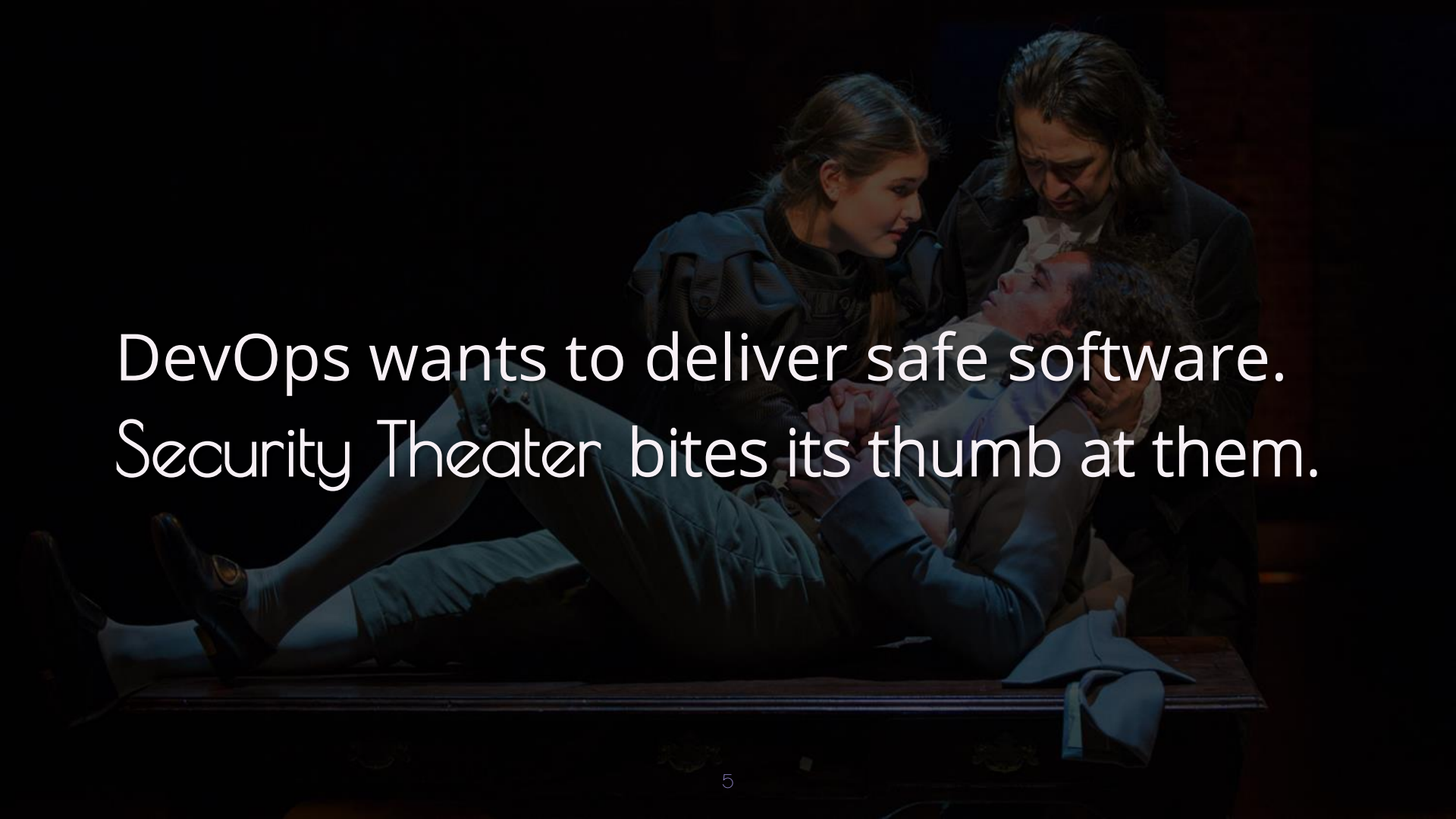
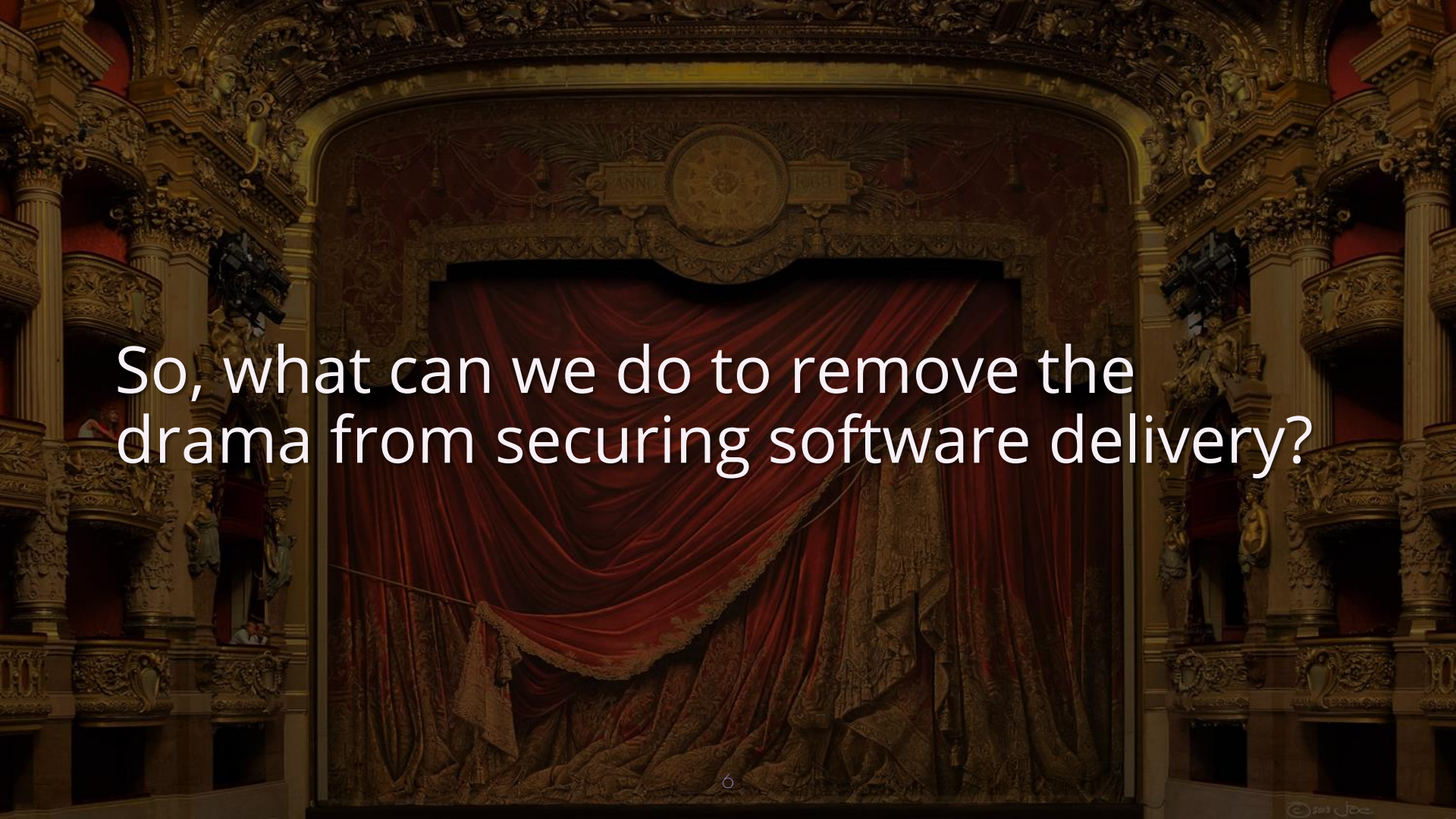

Exit Stage Left:
Eradicating Security Theater
Kelly Shortridge (@swagitda_)
Snykcon 2020 Keynote


Hi, I'm Kelly
CAPSULE8


Information security sees itself as the underappreciated star on the I.T. stage


...infosec is certainly a primmadonna


DevOps wants to deliver safe software.
Security Theater bites its thumb at them.

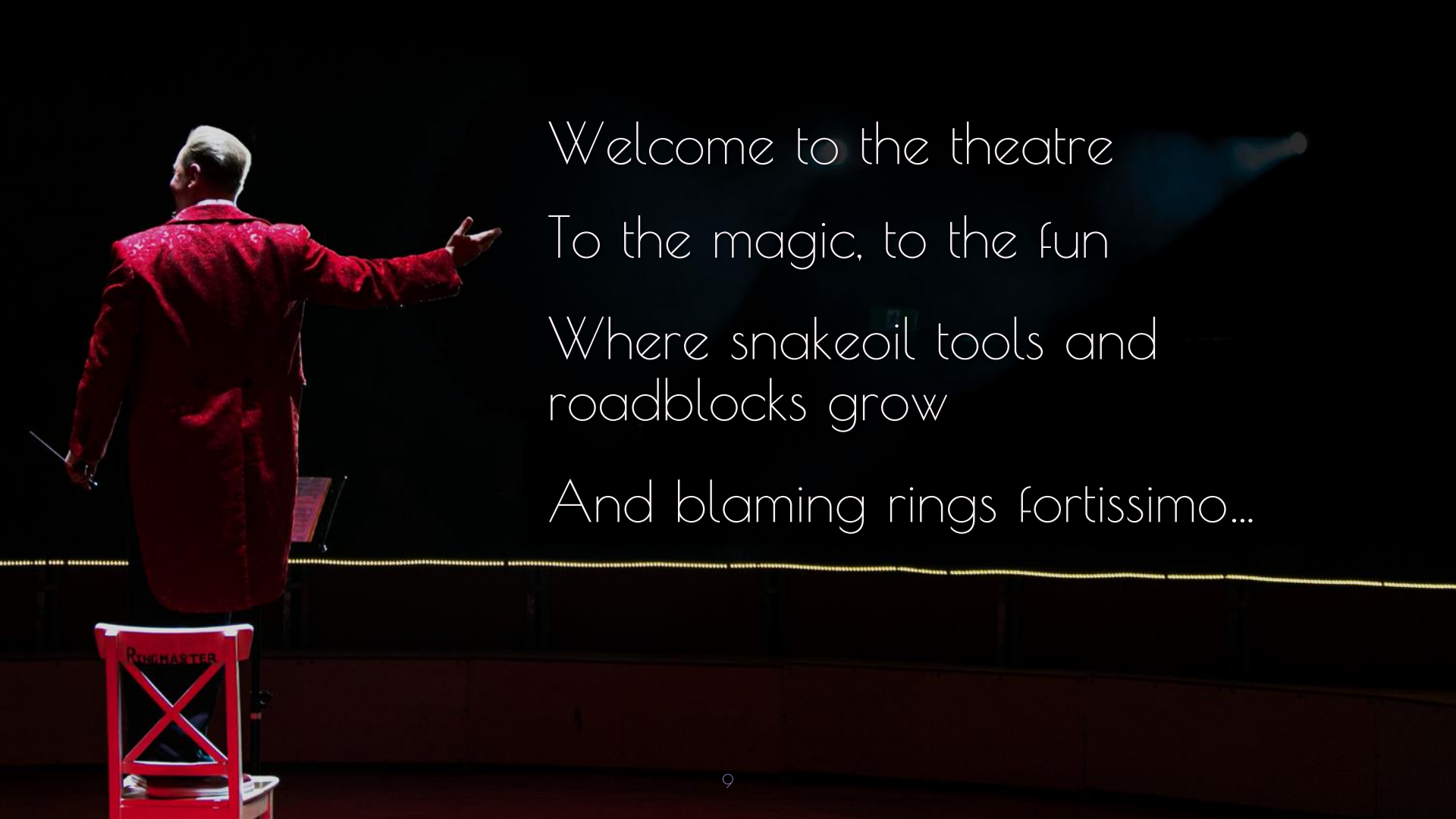

So, what can we do to remove the drama from securing software delivery?

Act I: Welcome to Security Theater

Act II: Fisticuffs

Act III: Redemption


Act 1: Welcome to
Security Theater



Welcome to the theatre

To the magic, to the fun

Where snakeoil tools and roadblocks grow

And blaming rings fortissimo...

Security Theater: producing the **perception** of improved security

Security Theater optimizes for **drama**

Processes apply to everyone just to catch the extra rare “bad apples”

“The strategy seems to be preventative control on everybody instead of damage control on those few.”

– Bjarte Bogsnes

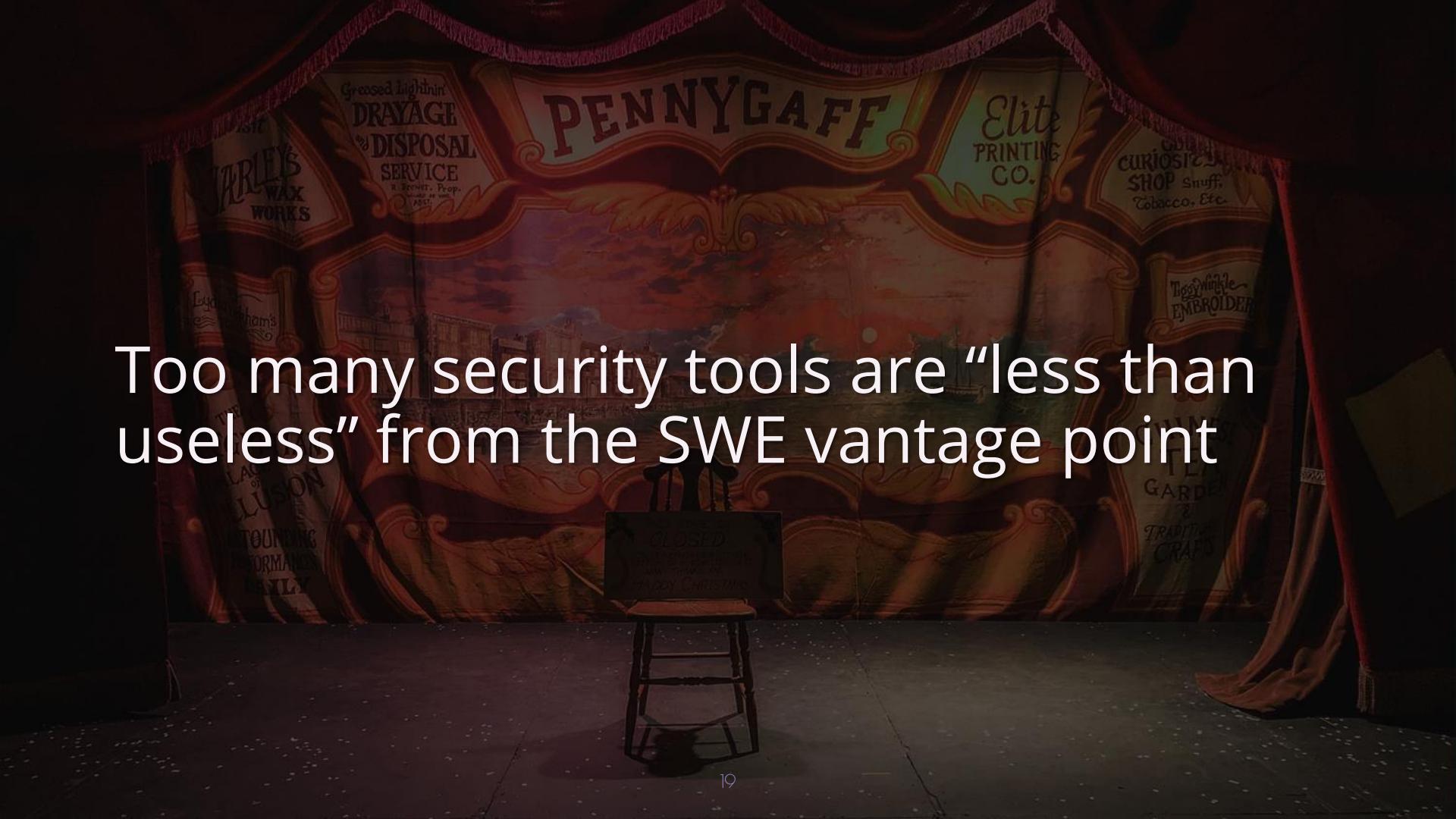
Hence: infosec as "Department of No"

This “makes life painful for the innocent but can be circumvented by the guilty”

Are scanner results going into a pretty report, or actually being fixed?


How long do your security tools take to run? Do they disrupt your pipelines?

…and are the security tools’ results worthy of attention in the first place?

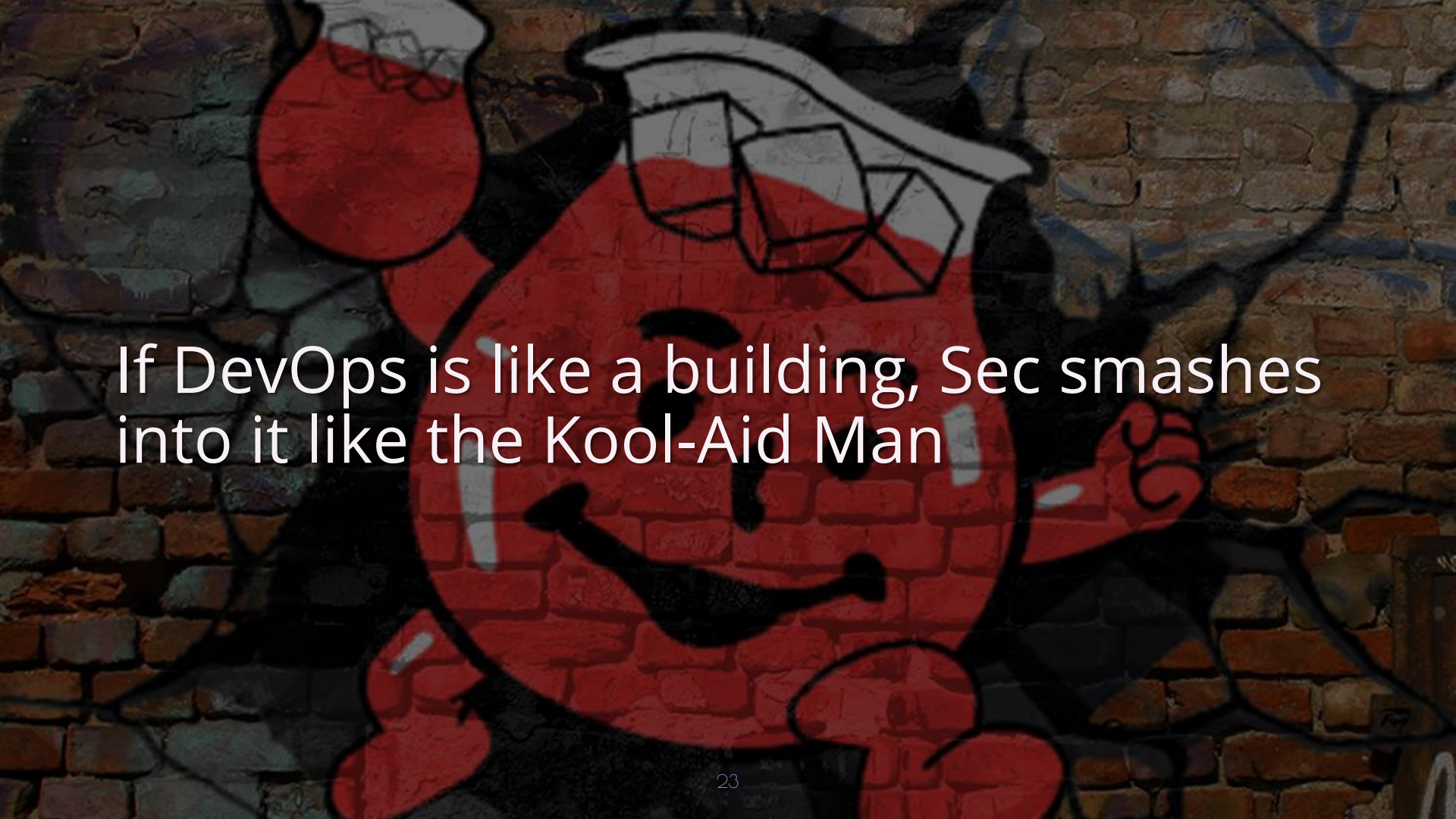


Too many security tools are “less than useless” from the SWE vantage point

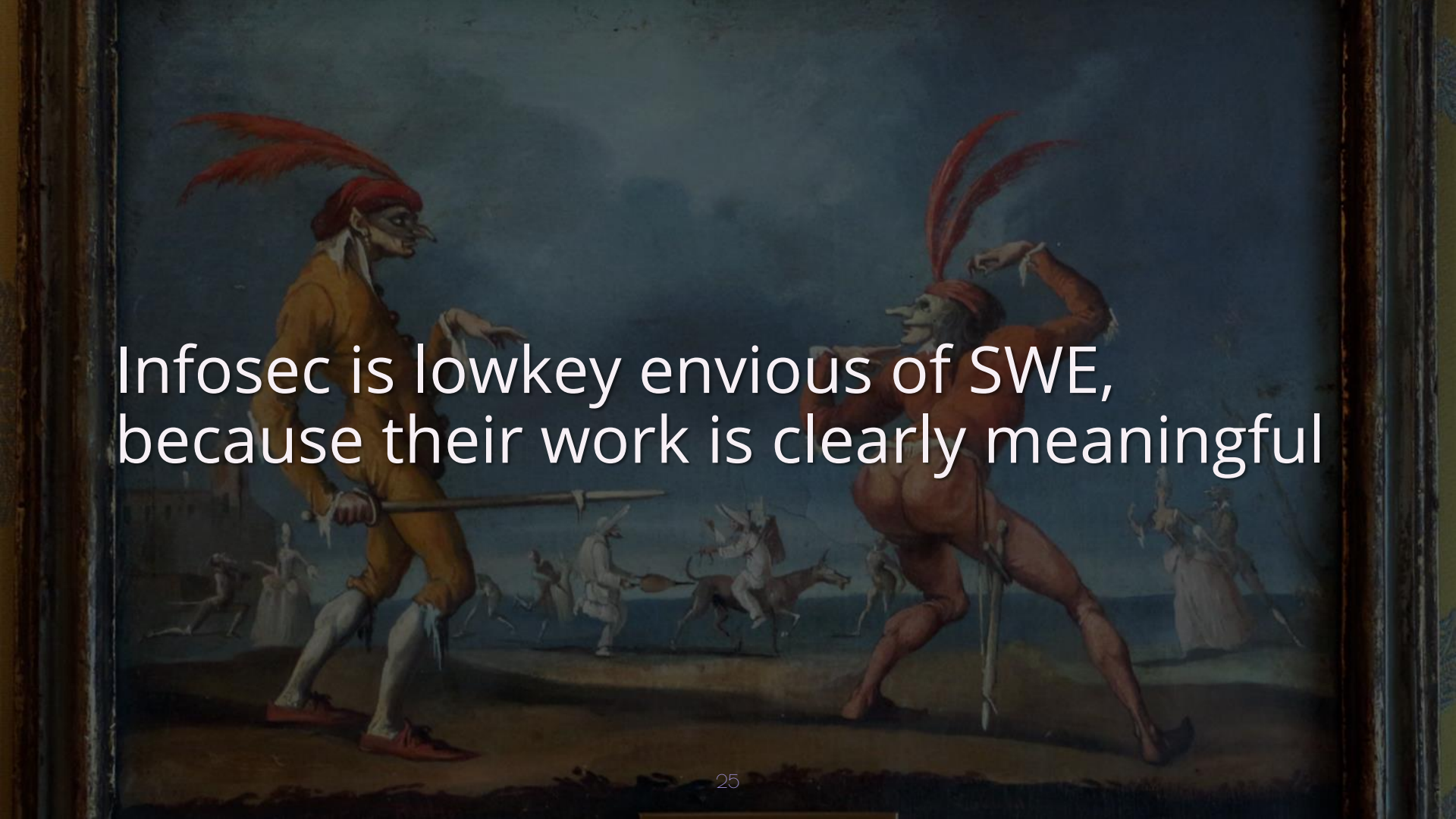
Shifting Left is often more “shift friction earlier” than “build in security by design”

DevSecOps: “I’m not a regular Security Theater, I’m a cool Security Theater”

Jamming security in is different than aligning accountability & responsibility

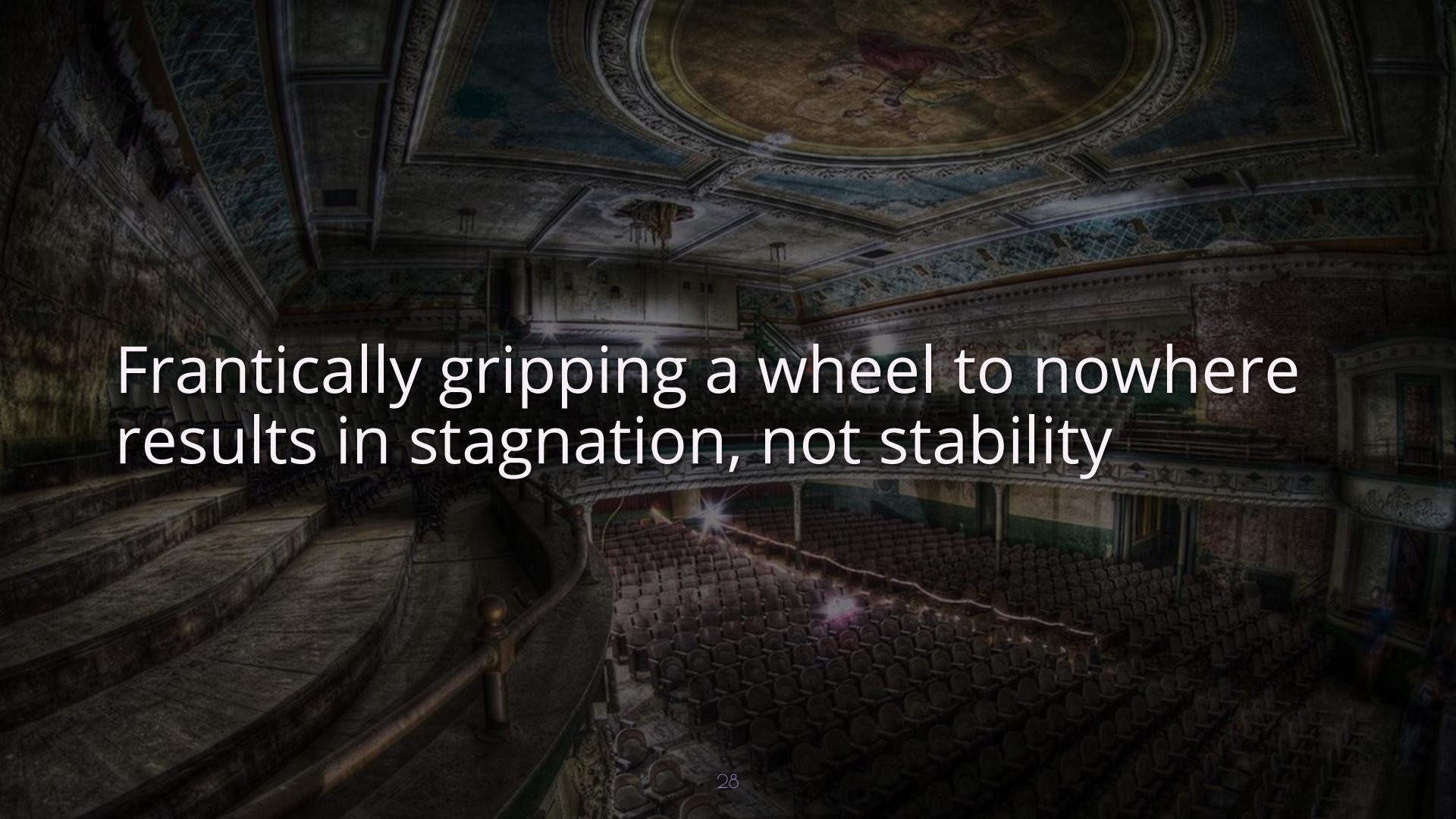

If DevOps is like a building, Sec smashes into it like the Kool-Aid Man

Driven by **FOMOsec**: wanting to *feel* like infosec is in control & not irrelevant


Infosec is lowkey envious of SWE,
because their work is clearly meaningful

FOMOsec's "Gotta Catch Em All" mindset is a classic at the Security Theater

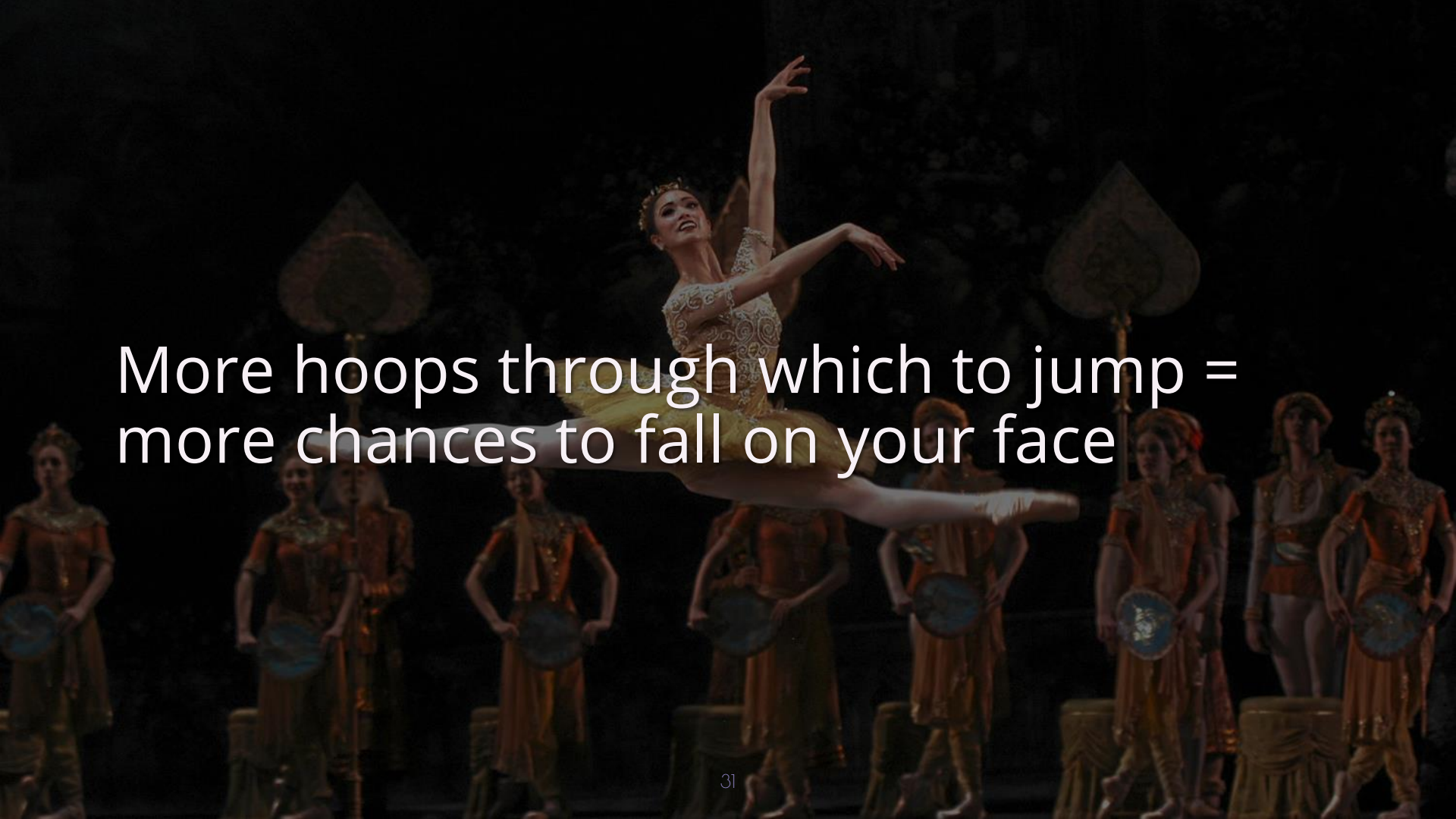
Infosec won’t sit at the Big Kids’ Business Table if it stays a Security Theater kid


Frantically gripping a wheel to nowhere results in stagnation, not stability

Stricter change management processes do not lead to greater stability

46% - 60% of changes by “conservative” orgs lead to degraded service

– State of DevOps research by Dr. Forsgren



More hoops through which to jump = more chances to fall on your face

Cumbersome change management will hinder speedy patch deployments, too

The reality: security must be adaptive

How do we spot Security Theater's red flags? And is there a better way ahead?


Act II: Fisticuffs

Fisticuffs emerges due to how to treat failure & where accountability rests

Security Chaos Engineering: Let’s harness failure to build knowledge

Security Theater: Avoid failure at all costs and punish any humans involved

Act II, Scene I:
The Duel

SCE: Failure is a natural part of systems

ST: Bad humans cause failures

SCE: Adapt to minimize incident impact

ST: Prevent failure from happening

SCE: Security is collaborative & open

ST: Security teams operate in a silo

SCE: Rewards system-level improvement

ST: Rewards rigidity & saying “no”

SCE Culture: Learning & experimenting

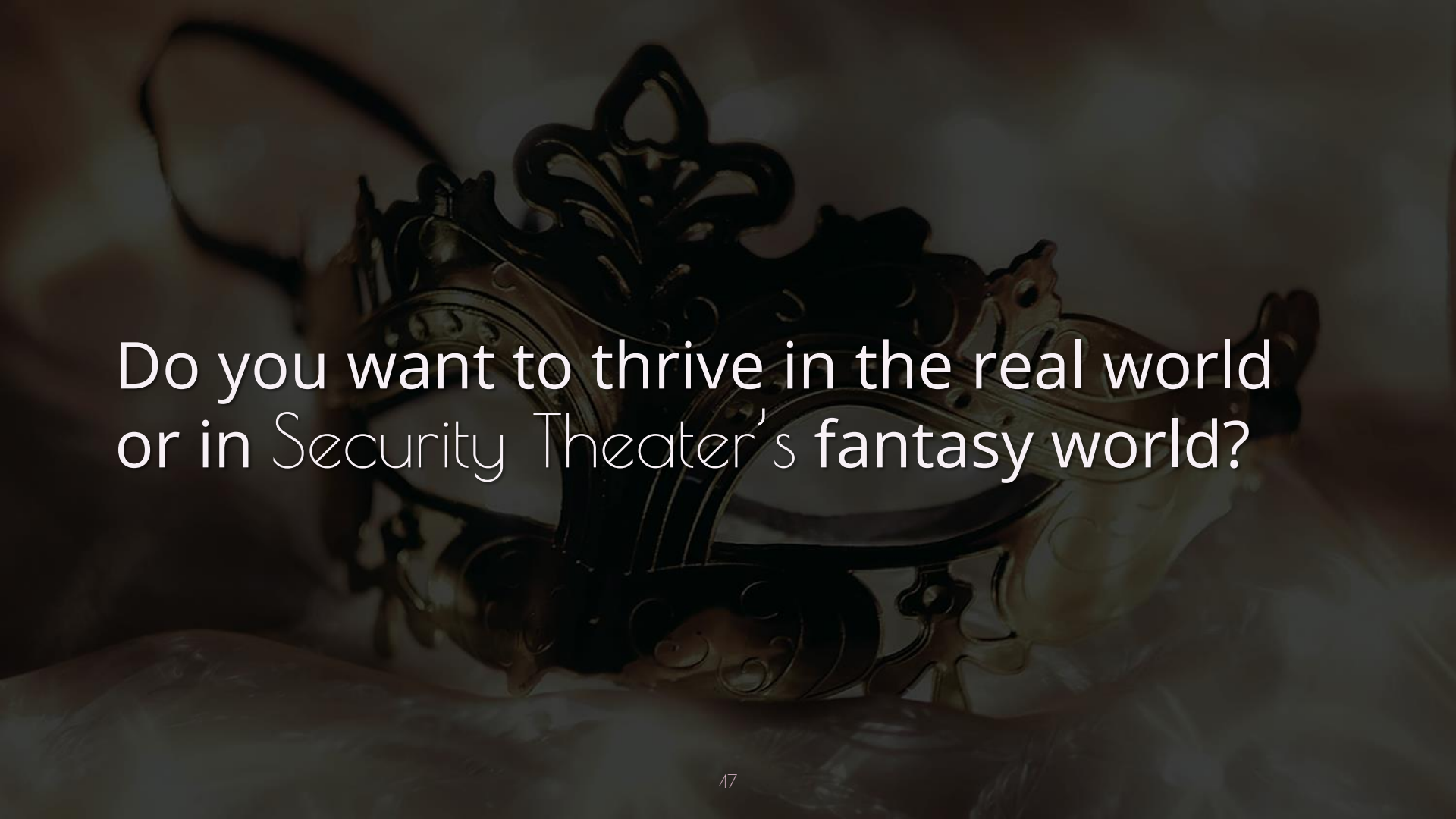
ST Culture: Fear and mistrust

SCE: Principles-based and adaptive

ST: Rule-based & prefers the status-quo

SCE Testing: Speedy and transparent

ST Testing: Manual, slow, and opaque


Do you want to thrive in the real world
or in Security Theater's fantasy world?

Security Chaos Engineering cares about meaningful outcomes (anti-FOMOSec)

SCE aligns with how dev and ops already thinks and operates

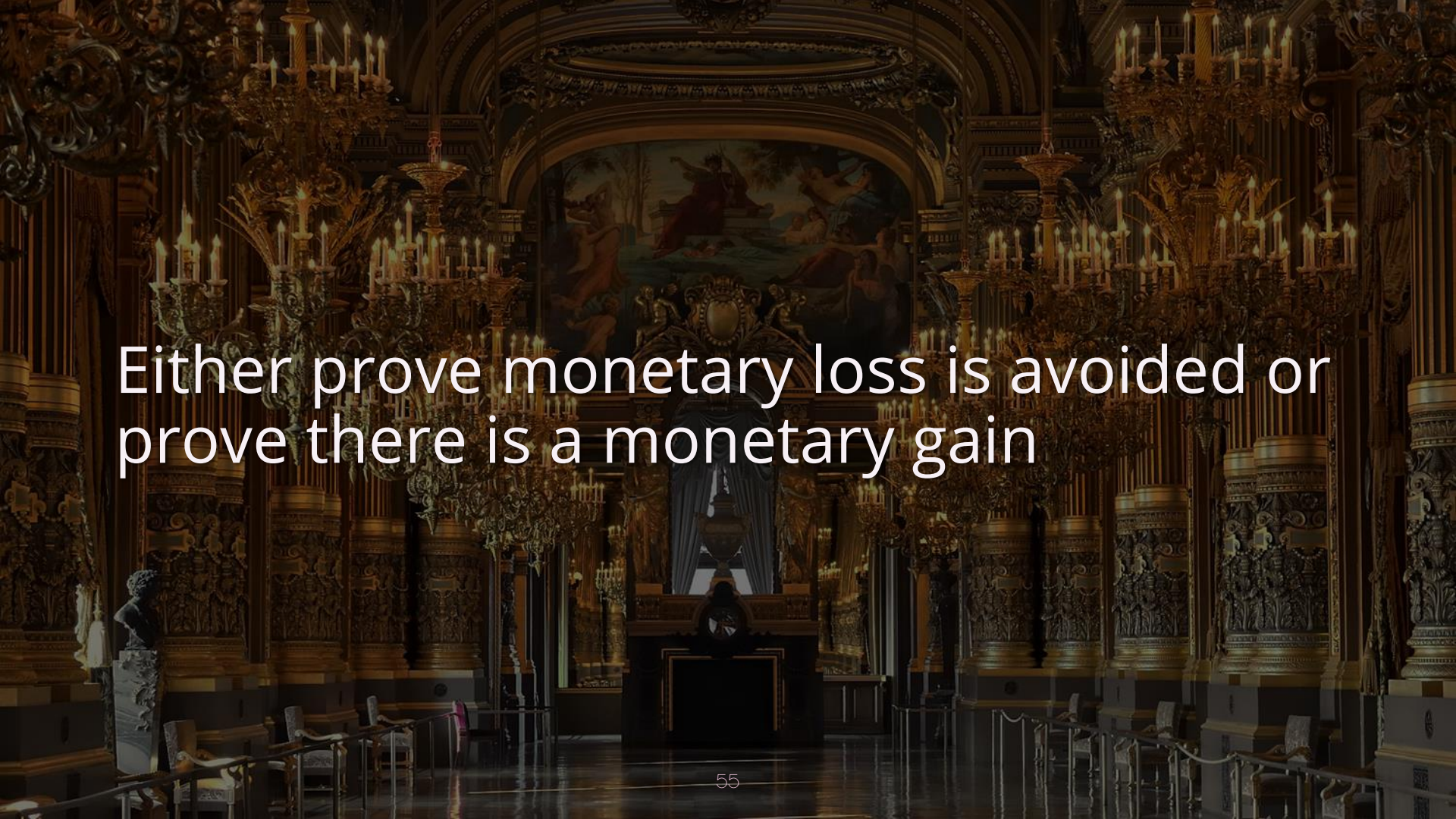
Joining the Security Theater results in a dangerous, self-fulfilling prophecy...


Strict control within a culture of fear
turns the innocent into bad apples


Act II, Scene II:
Judgment

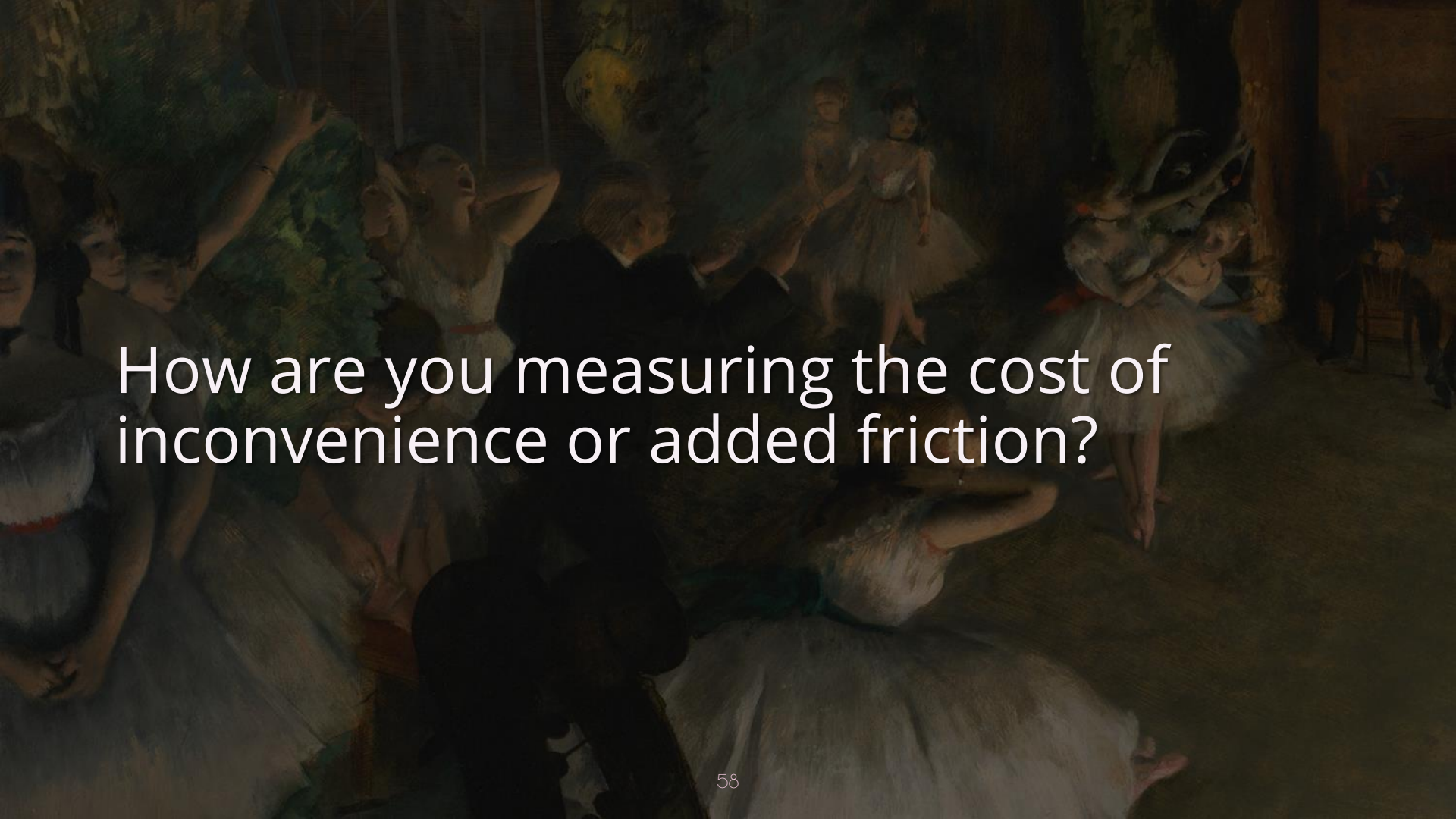

Security Theatre shuns fair judgment

Moving away from ST towards SCE is a move towards success measurement


Either prove monetary loss is avoided or prove there is a monetary gain

What are the negative externalities of your security program?

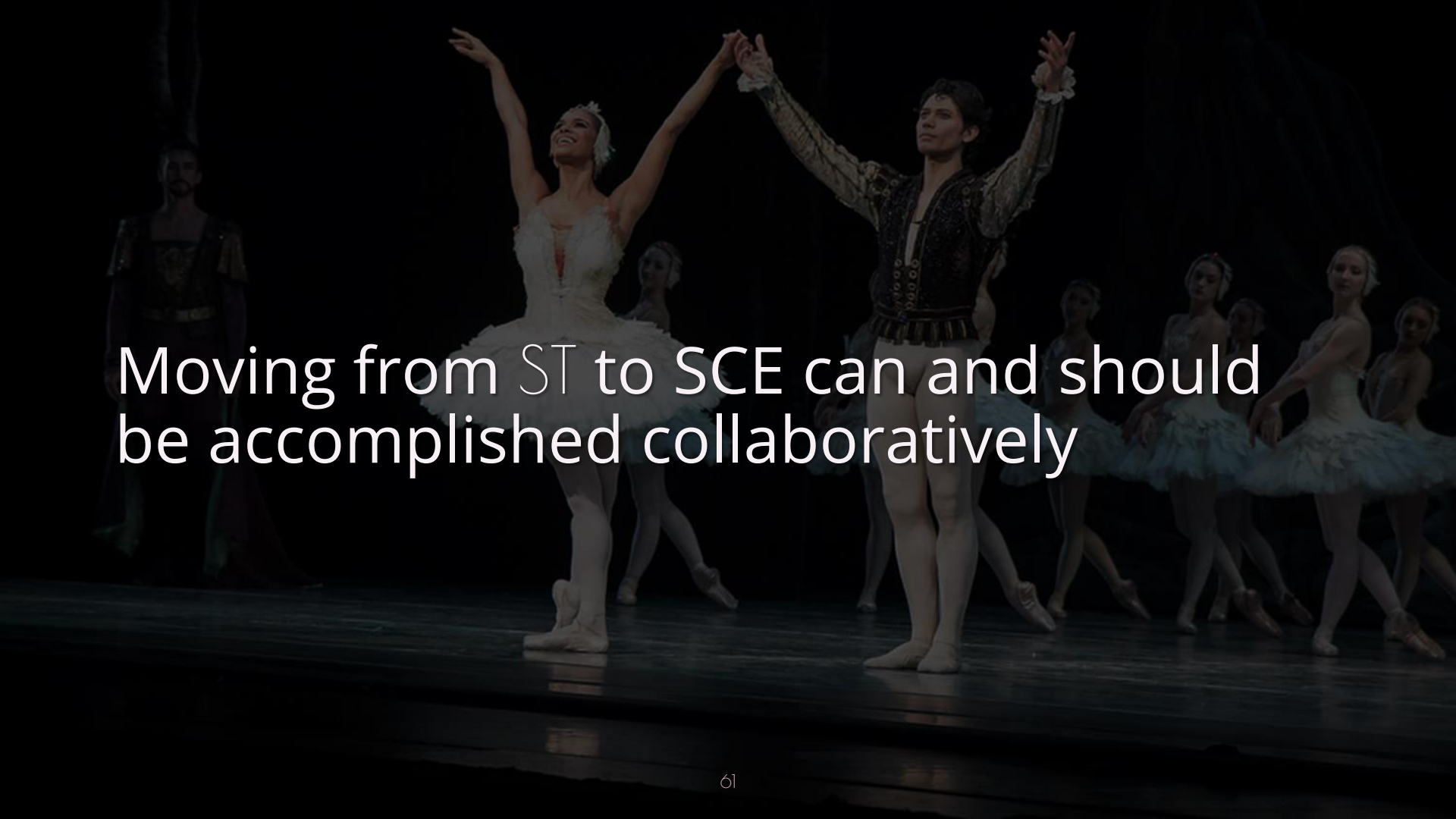
Security metrics are too often tone deaf to what the rest of the org is doing



How are you measuring the cost of inconvenience or added friction?

Compare security code review coverage vs. lead time or deploy frequency

Investigate any sources of friction and validate that it isn’t due to security



Moving from ST to SCE can and should be accomplished collaboratively


Act III: Redemption

Security approvals don’t have to suck

Optimize for “just enough” in security reviews and use evidence, not opinions

Little’s Law: $L = \lambda W$

Reducing the security review queue size reduces the lead time to deploy

Treat high-impact changes differently than low-impact changes

Act III, Scene I:
High-risk, high-impact changes

High-risk, high impact changes are worthy of security team scrutiny

High-risk, high-impact (HRHI) = affects the whole org or multiple products

Ensures the security team’s efforts are prioritized more effectively

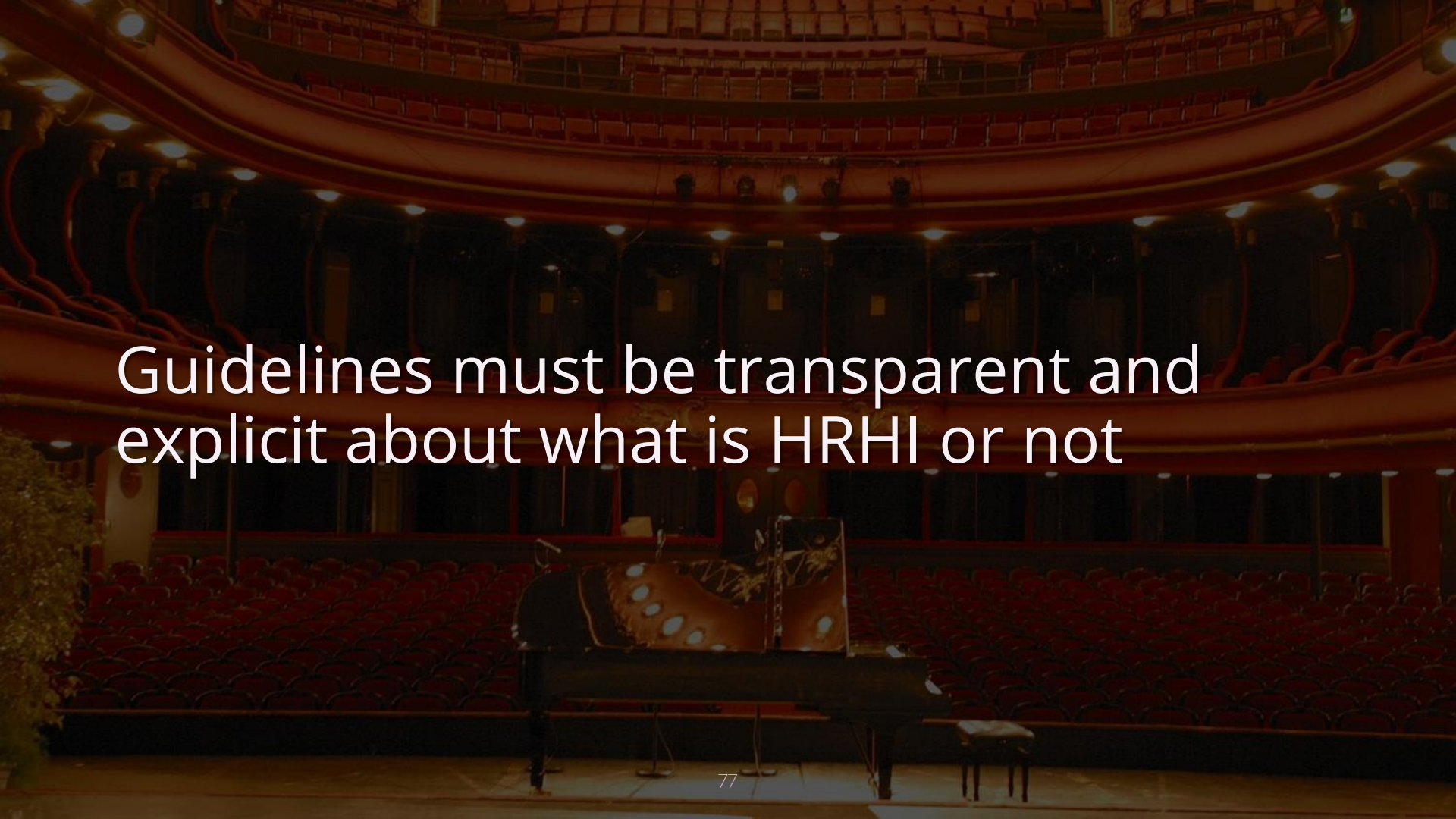

Who *doesn't* want to focus on higher-value work rather than battling noise?

Product team is responsible for creating awareness & discussion of changes

HRHI changes require sign-off by P&E leaders & acceptance of accountability

Security reviews are timeboxed & limited in number to optimize attention

Build a redundancy plan for extra urgent HRHI changes (just in case)



Guidelines must be transparent and explicit about what is HRHI or not

Retros on change impacts build knowledge & should be shared widely

Act III, Scene II:
Low-risk, low-impact changes

tl;dr straightforward classification and automated exemption process


Incentivize appropriate security handling vs. incentivizing bypasses

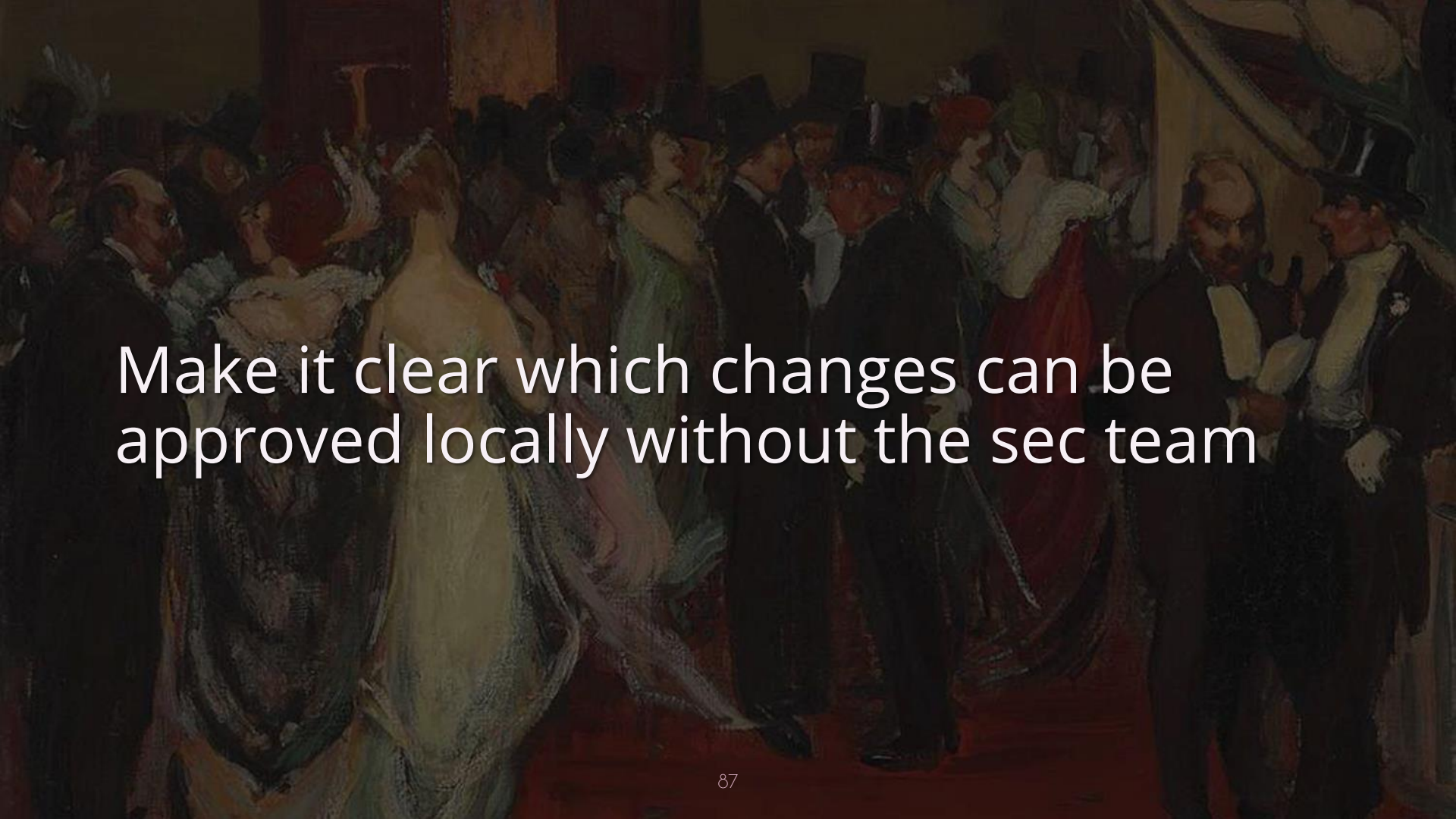
Assessments & cross-team discussions are unnecessary for LRLI changes

Classification with recommended steps for low, standard, critical, informational

Pipeline automation can facilitate exception handling (but also document!)

Encourage local approvals and peer reviews for LRLI changes

Track if LRLI changes are resulting in high impacts, then tweak the process



Make it clear which changes can be approved locally without the sec team

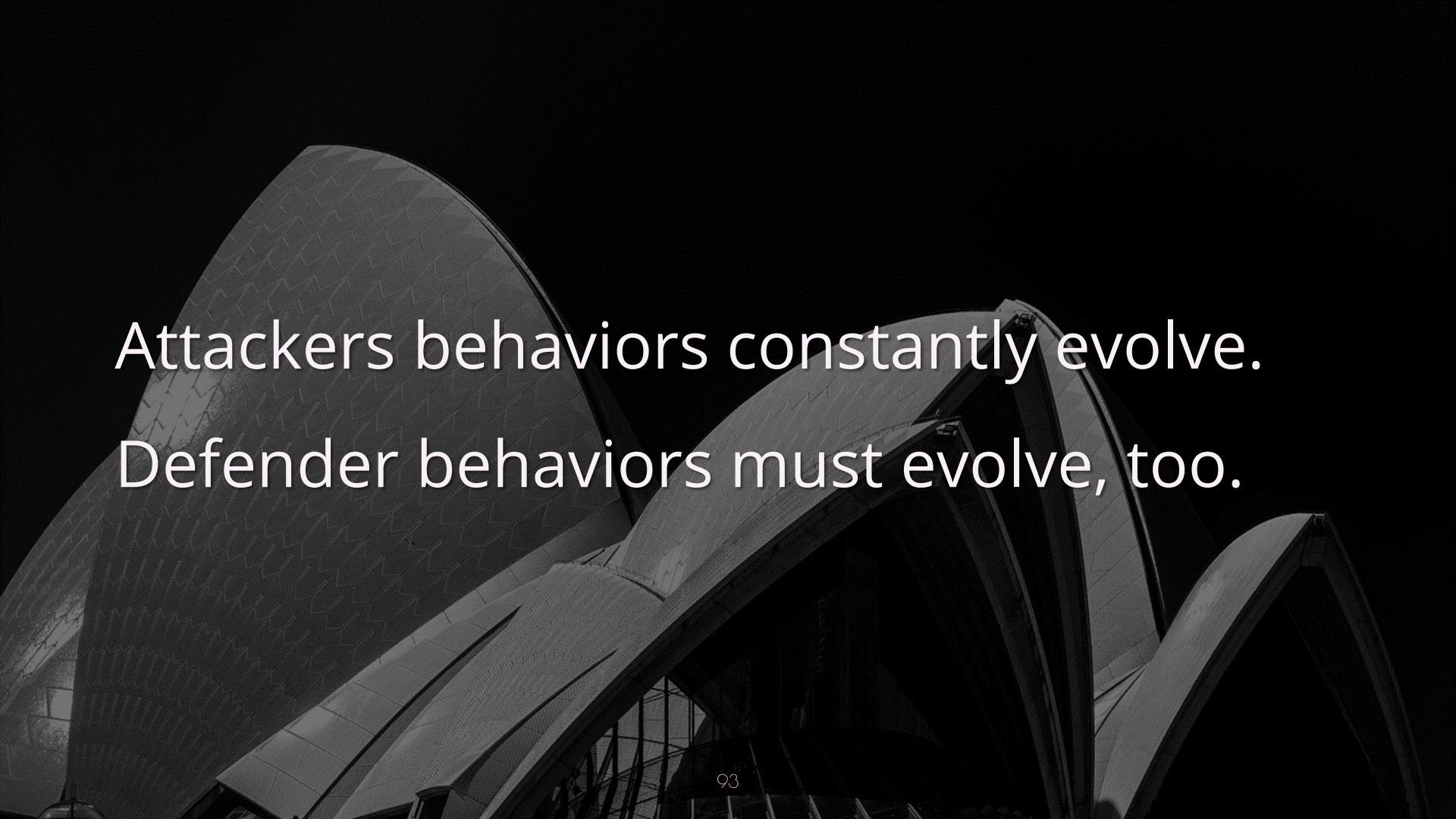
LRLI changes could include pricing updates, new cat gifs, tooltip copy...


The Grande Finale

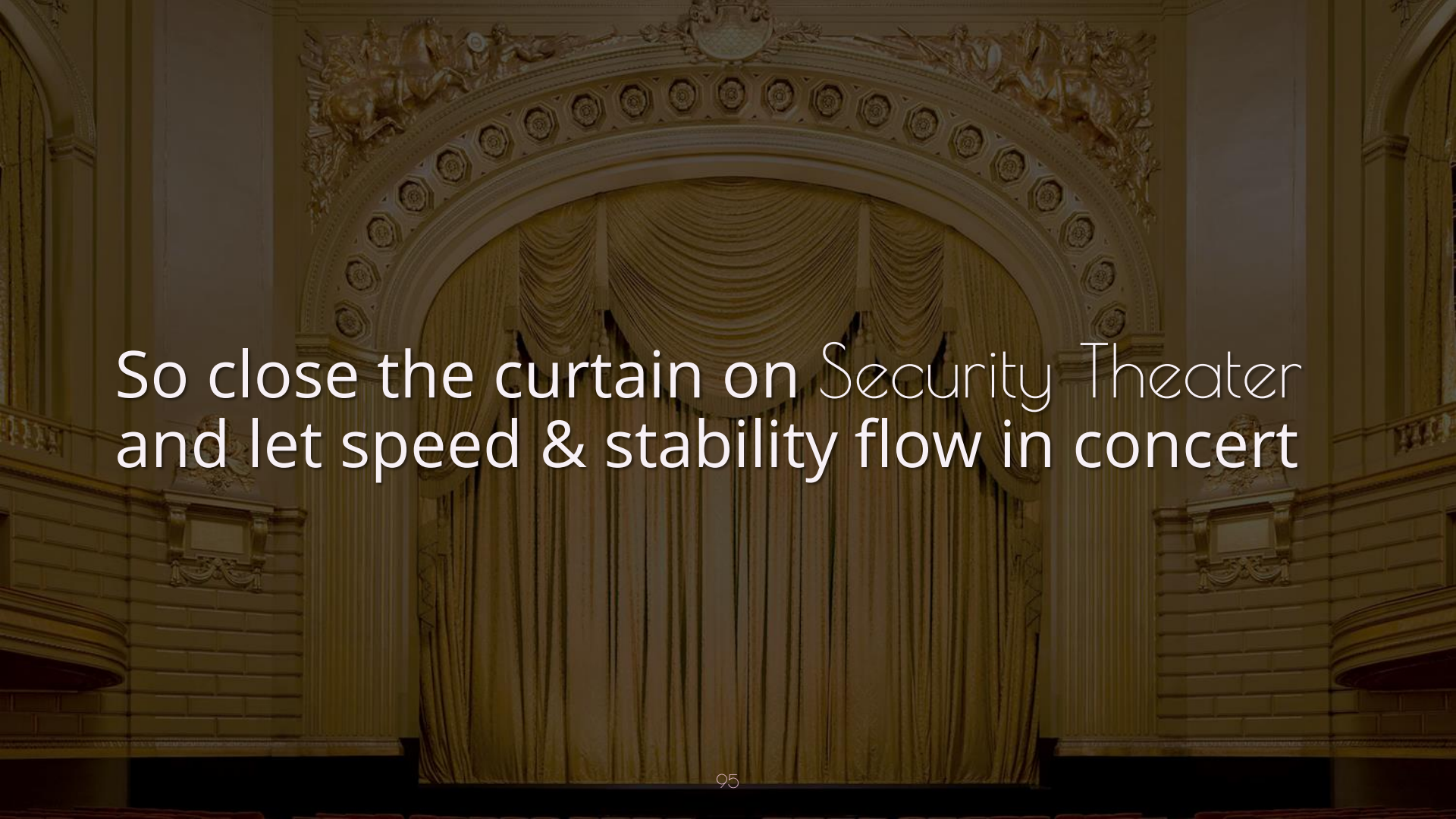

Security Theater prioritizes gatekeeping
more than security outcomes

Heavy security approvals promote friction and bottlenecks, not stability

Strive for continuous improvement through Security Chaos Engineering


Attackers behaviors constantly evolve.
Defender behaviors must evolve, too.

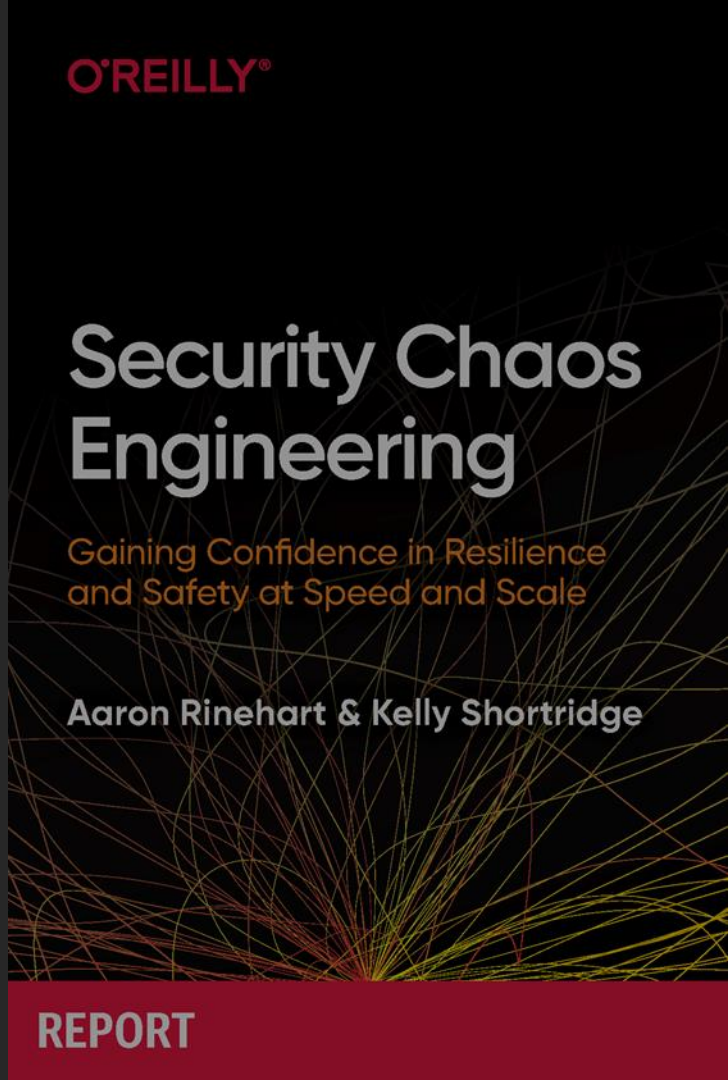

Treat security teams as advisors & hold
P&E teams accountable for changes


So close the curtain on Security Theater
and let speed & stability flow in concert

“People don't want their lives fixed. Nobody wants their problems solved. Their dramas. Their distractions. Their stories resolved. Their messes cleaned up. Because what would they have left? Just the big scary unknown.”

– Chuck Palahniuk

Our forthcoming report (early November): https://bit.ly/35fYxNR



@swagitda_

/in/kellyshortridge

https://get.oreilly.com/ind_security-chaos-engineering.html