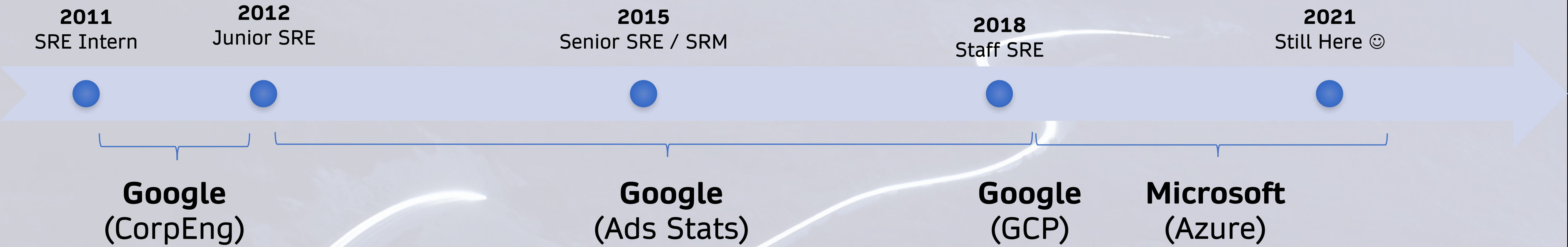
10 LESSONS LEARNED IN 10 YEARS OF SRE

Andrea Spadaccini, Ph.D.

Principal Software Engineer – SRE @ Microsoft Azure

SRECON’21 – 12-14 October 2021

My SRE Journey



2011
SRE Intern

2012
Junior SRE

2015
Senior SRE / SRM

2018
Staff SRE

2021
Still Here ☺

Google
(CorpEng)

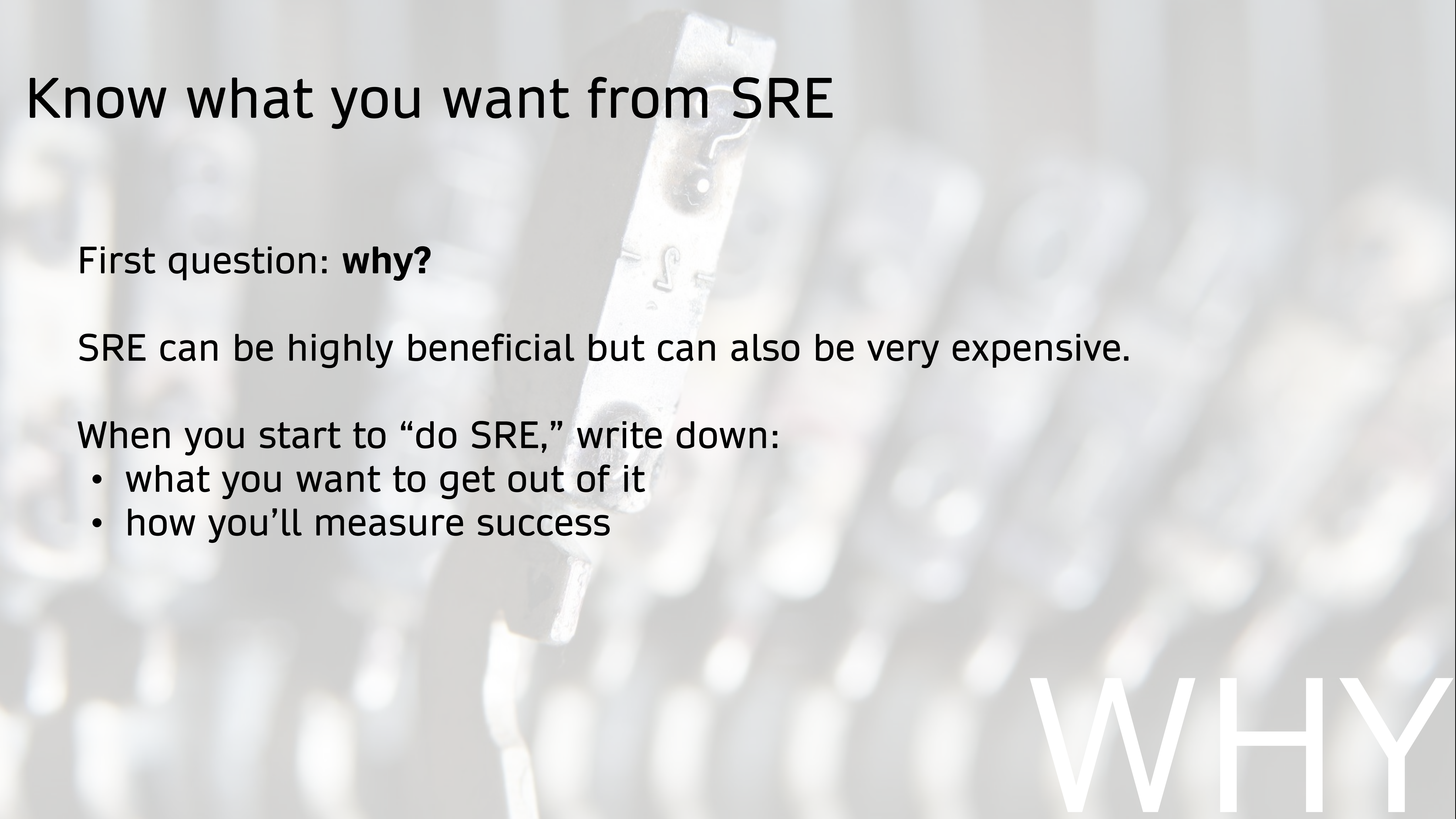
Google
(Ads Stats)

Google
(GCP)

Microsoft
(Azure)

Starting SRE

Know what you want from SRE



First question: **why?**

SRE can be highly beneficial but can also be very expensive.

When you start to “do SRE,” write down:

- what you want to get out of it
- how you’ll measure success

WHY

Align with business goals / customer needs

SRE must serve business goals

Anti-pattern: SRE roadmap drifts from product dev roadmap

Best practice: establish feedback loops

Expertise matters

Sometimes, generalists are not sufficient

Best practice: seed teams with T-shaped experts

DEPTH

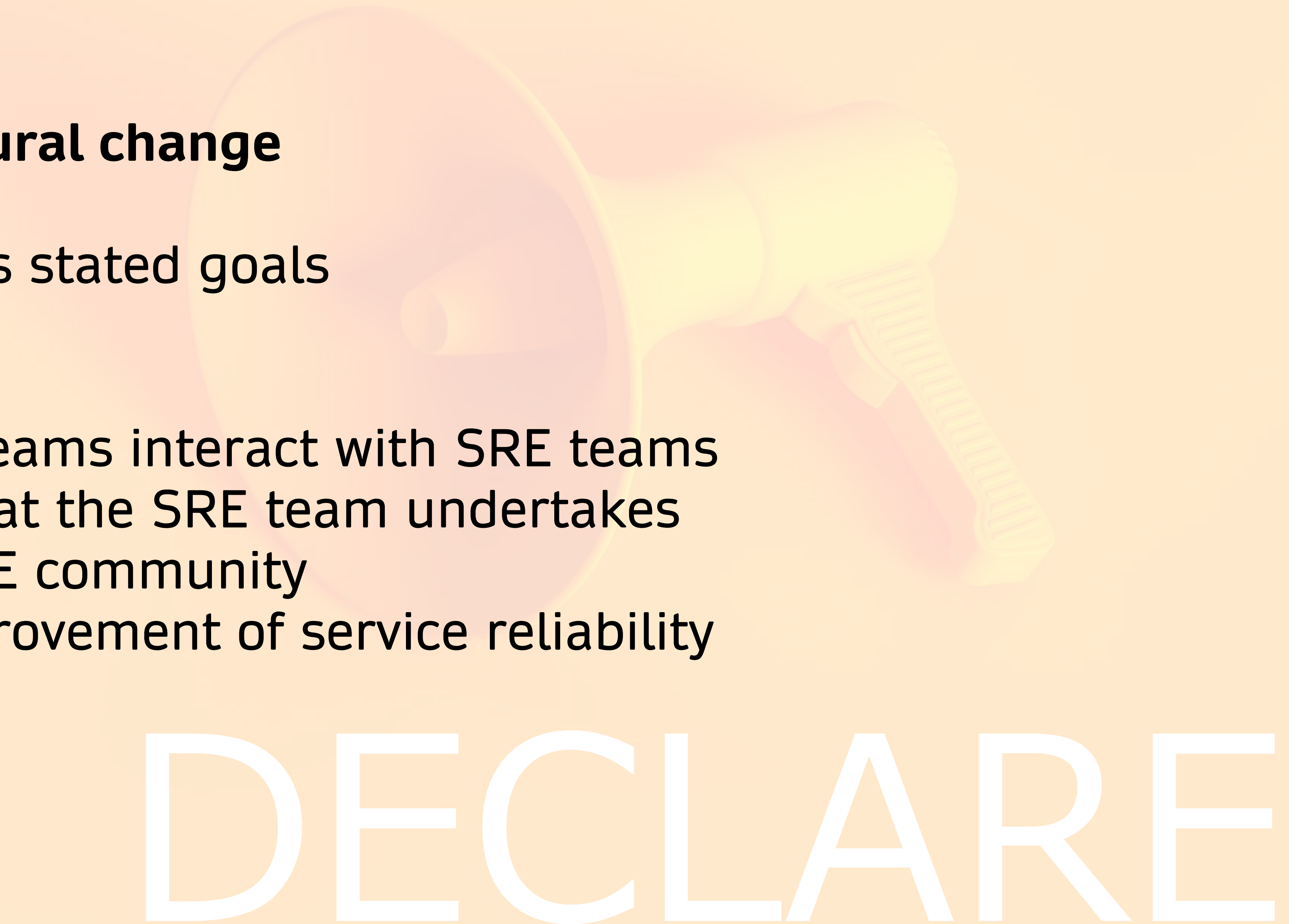
You cannot declare “SRE”

Introducing SRE is a **cultural change**

Measure progress towards stated goals

Other proxies of success

- How the product dev teams interact with SRE teams
- The type of projects that the SRE team undertakes
- The presence of an SRE community
- Measurement and improvement of service reliability



Build trust

SRE relies on **shared ownership**

Trust is **vital**

Levels of trust

- Executives
- Engineering managers / senior SREs
- Engineers

Ways to establish trust

- Alignment and advocacy through continuous bidirectional feedback loops
- Delivery of complete, incrementally more impactful projects over time

Steady-state SRE

Finish what you start

It is **hard** to stay focused

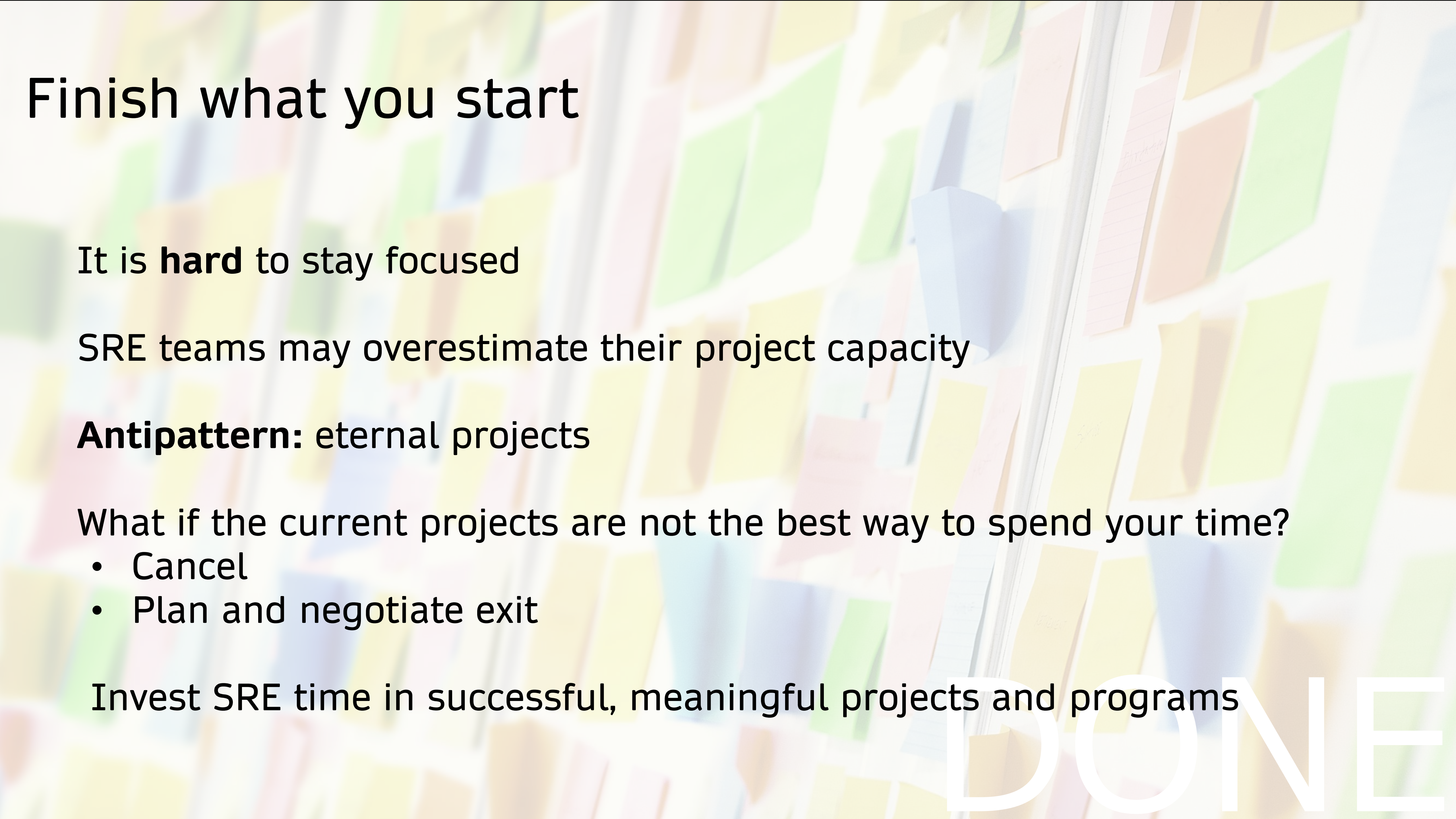
SRE teams may overestimate their project capacity

Antipattern: eternal projects

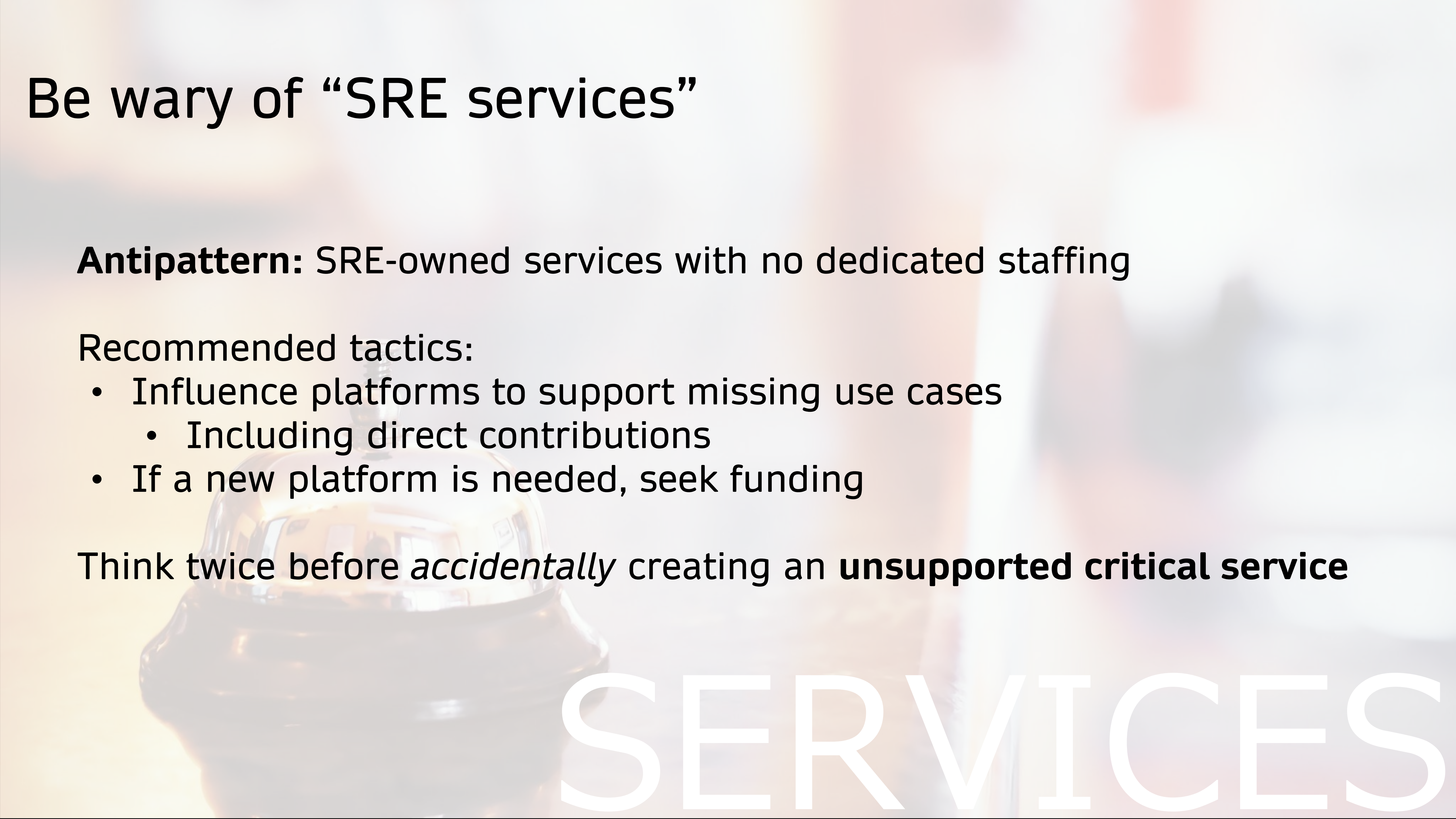
What if the current projects are not the best way to spend your time?

- Cancel
- Plan and negotiate exit

Invest SRE time in successful, meaningful projects and programs



Be wary of “SRE services”



Antipattern: SRE-owned services with no dedicated staffing

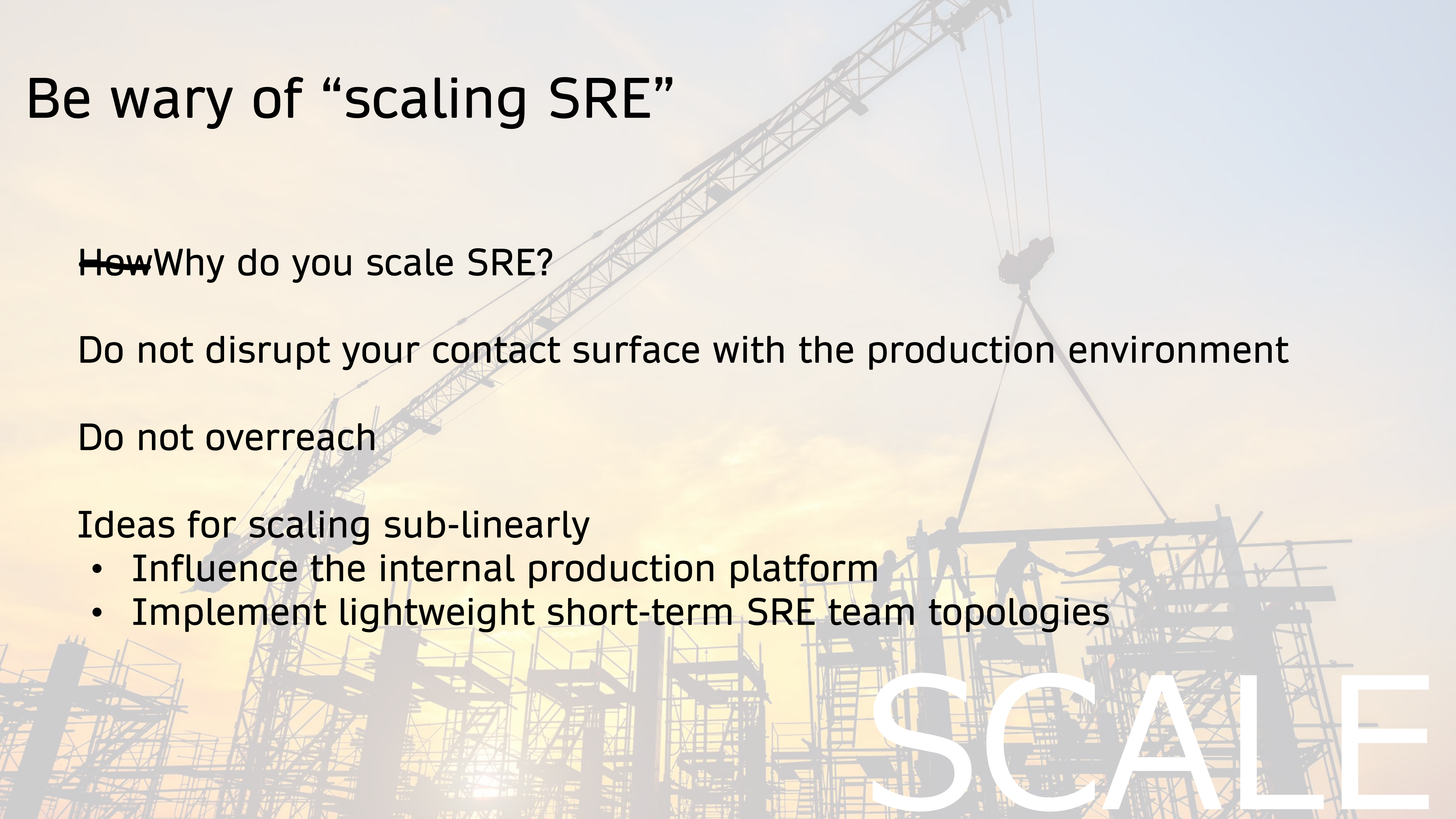
Recommended tactics:

- Influence platforms to support missing use cases
  - Including direct contributions
- If a new platform is needed, seek funding

Think twice before *accidentally* creating an **unsupported critical service**

SERVICES

Be wary of “scaling SRE”



~~How~~Why do you scale SRE?

Do not disrupt your contact surface with the production environment

Do not overreach

Ideas for scaling sub-linearly

- Influence the internal production platform
- Implement lightweight short-term SRE team topologies

SCALE

Stay humble

SREs, by necessity, are excellent engineers

SREs, by trade, develop key knowledge and skills

SREs, by design, work across teams

Therefore SRE **may** develop a tiny bit of hubris and arrogance ☺

It is essential that SREs stay humble

Keep asking questions with respect, an open mind and positive attitude

HUMILITY

Talk more

Our profession relies on collaboration

Share problems and solutions within SRE

Invest in your network

Just talk more and have fun ☺

TALK

THANKS!

Andrea Spadaccini

andrea.spadaccini@{gmail,microsoft}.com

Twitter / GitHub: lupino3