PlayStation

Sony PlayStation Network – Approach to Site Reliability Engineering

Yu Suefuji / Yuya Miyahara

2023/6/14v 1.0

Sony
Interactive
Entertainment

About us

Sony Interactive Entertainment

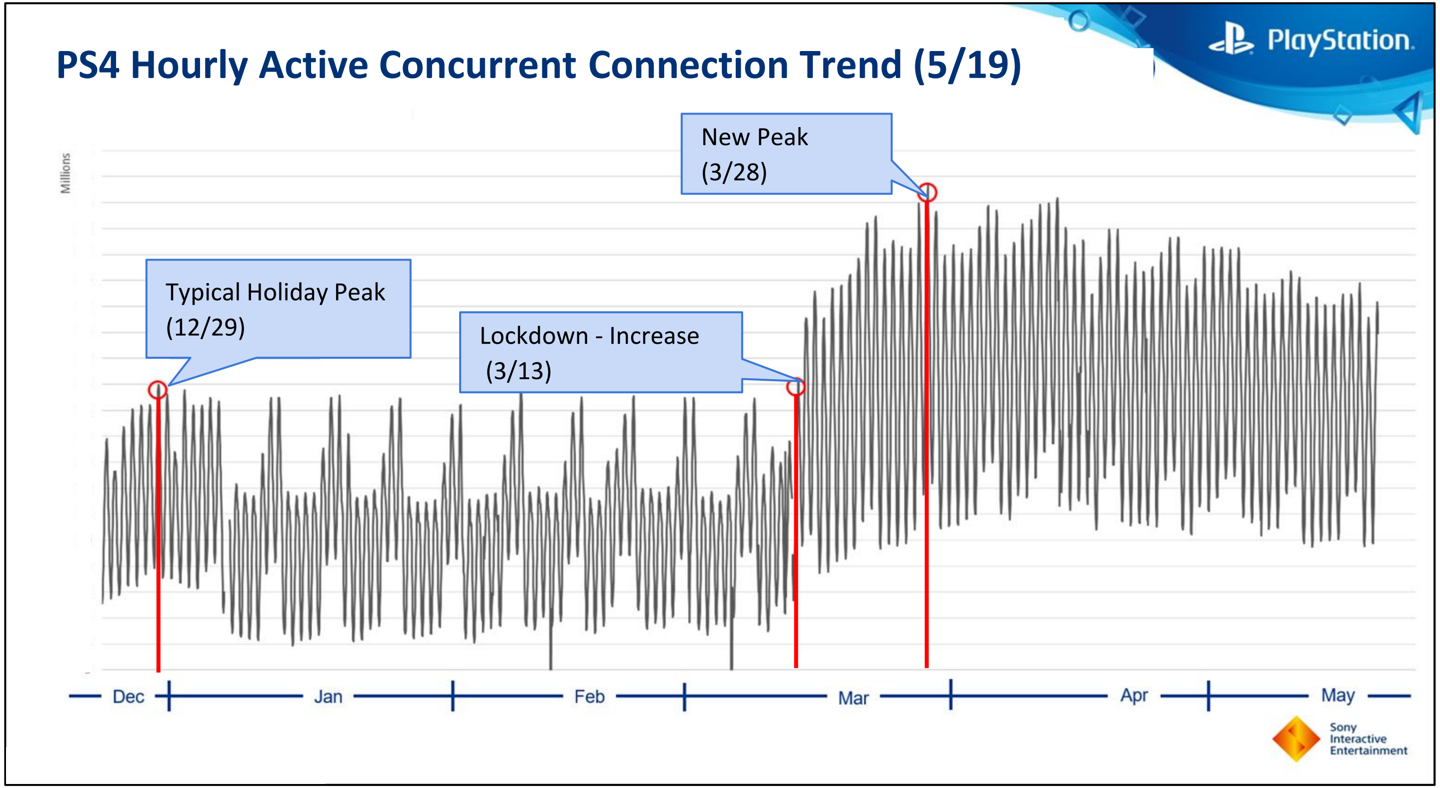
1994	2000	2006	2013	2016	2020	2023
PlayStation®	PlayStation®2	PlayStation®3	PlayStation®4	PlayStation®4 Pro PlayStation®VR	PlayStation®5	PlayStation®VR2

Download
NW add-on

Network Services

COVID-19 Lockdown

Inside report 2020-05



PS4 Hourly Active Concurrent Connection Trend (5/19)

Typical Holiday Peak (12/29)

Lockdown - Increase (3/13)

New Peak (3/28)

#PS5Launch #Singapore #GardenBytheBays

#PS5Launch #Japan #神田明神 #Tokyo

#PS5Launch #SaudiArabia #IthraTower #Khobar

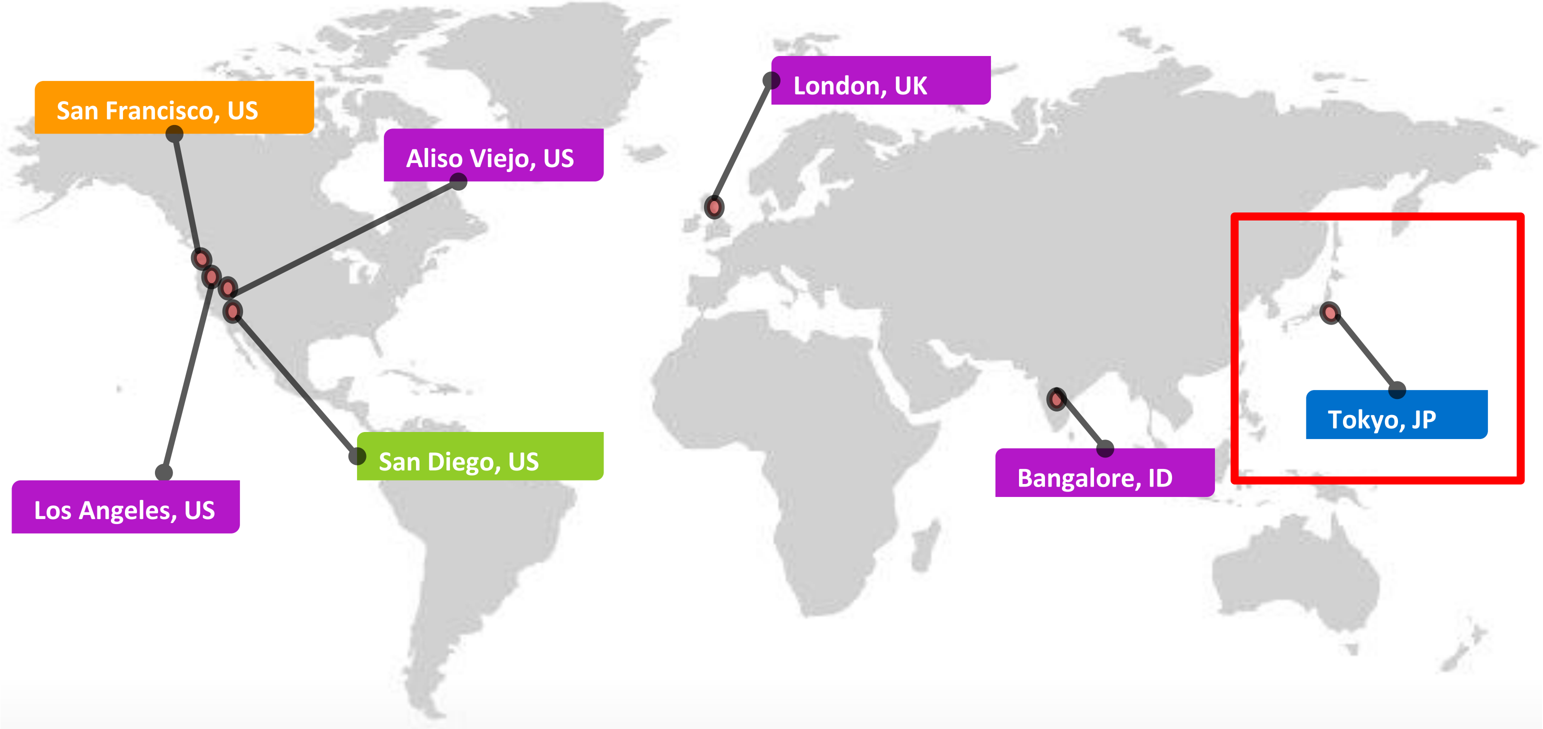
#PS5Launch #NewZealand #SkyTower #Auckland

PLAY HAS NO LIMITS™

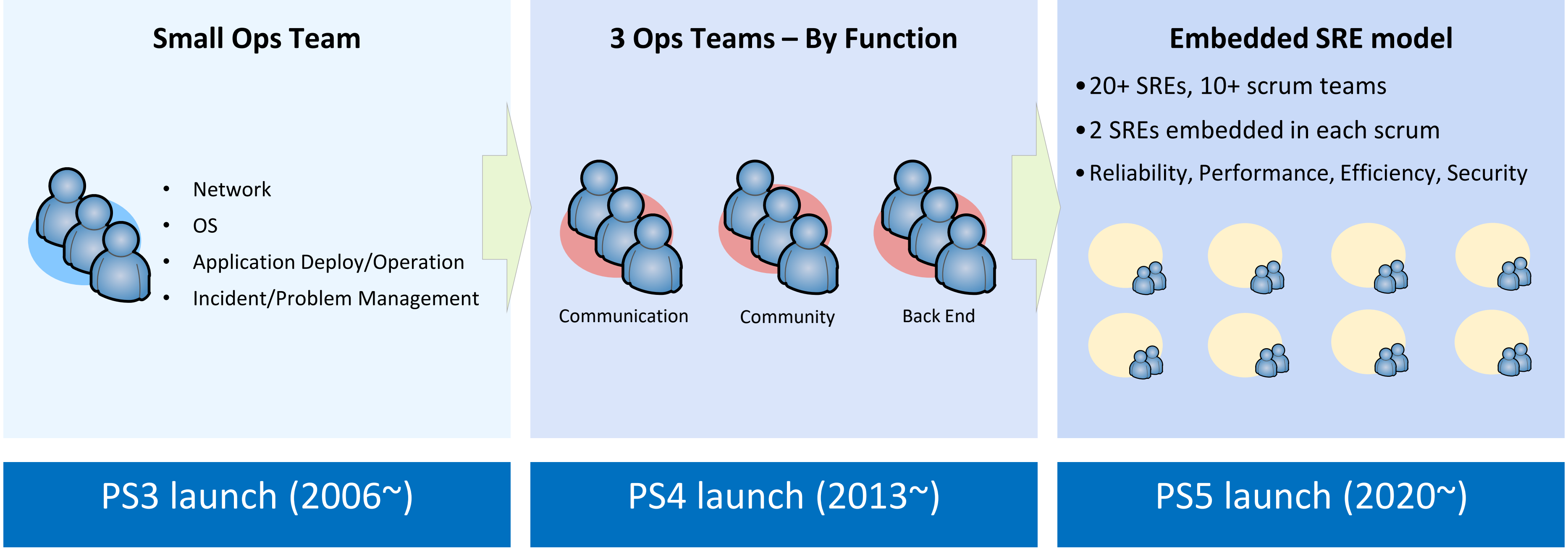
遊びの限界を超える

The history of Tokyo SRE team

PlayStation Network Server team map



The history of Tokyo SRE team



Small Ops Team

- Network
- OS
- Application Deploy/Operation
- Incident/Problem Management

PS3 launch (2006~)

3 Ops Teams – By Function

PS4 launch (2013~)

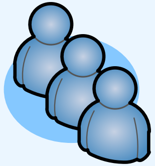
Embedded SRE model

- 20+ SREs, 10+ scrum teams
- 2 SREs embedded in each scrum
- Reliability, Performance, Efficiency, Security

PS5 launch (2020~)

The history of Tokyo SRE team

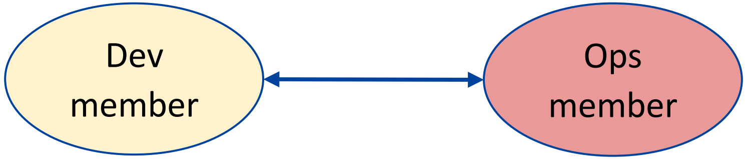
Small Ops Team



- Network
- OS
- Application Deploy/Operation
- Incident/Problem Management

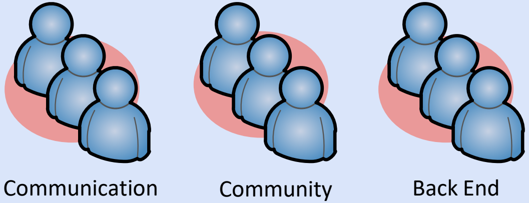
PS3 launch (2006~)

- PlayStation Network launched in 2006 alongside PS3
  - **Application deployment and operations for all services by 1 person** **(Sign-In, Messaging, Profile, Friend, Ranking, Matchmaking, etc.)**
- Direct Individual Communications between Dev and Ops
- 24/7 monitoring by external contractors
- Team expanded as PlayStation Network users increased



The history of Tokyo SRE team

3 Ops Teams – By Function

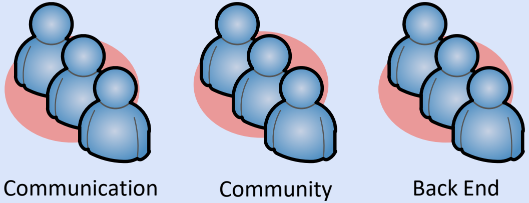


PS4 launch (2013~)

- PS4 launched in 2013, bringing big changes to network services
- **Infrastructure**
  - On-premise to Amazon Web Services
- **System architecture**
  - Sign-In, Authentication, Notification, Profile, Presence, etc.
- **Network interconnectivity and dependencies**
  - Tokyo, San Diego, San Francisco
- **Global service operations in San Diego**
  - 24/7 monitoring from external vendor to internal team

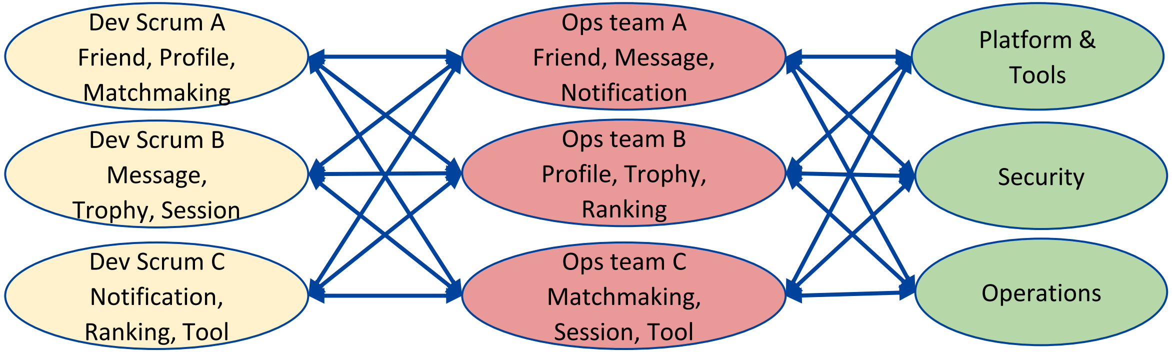
The history of Tokyo SRE team

3 Ops Teams – By Function



PS4 launch (2013~)

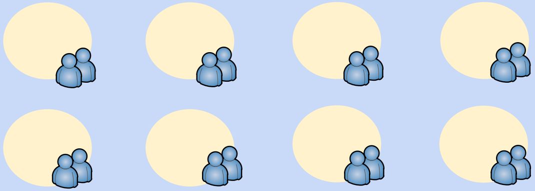
- Dev and Ops team expanded independently
  - **Each Ops team had to communicate with each Dev scrum since their responsibilities were different**
- As services grew, Dev<->Ops communications became more important
- At the same time, # of communication channels became unmanageable



The history of Tokyo SRE team

Embedded SRE model

- 20+ SREs, 10+ scrum teams
- 2 SREs embedded in each scrum
- Reliability, Performance, Efficiency, Security



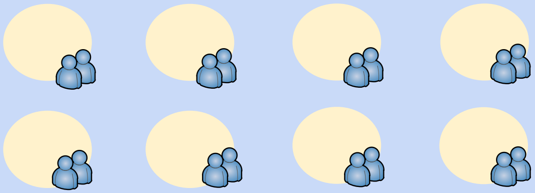
PS5 launch (2020~)

- PS5 launched in 2020
- Team transformed from traditional Dev/Ops to Embedded SRE model
  - **Joined scrum activities (Daily, sprint planning, etc.)**
  - **Managed Ops tasks as a scrum**
- Workstyle and Culture Change
  - **Different roles between Dev and Ops**
  - **Dev gained understanding of Ops, and vice versa**
  - **Collaborated on deployment strategies, resource provisioning, monitoring, alerting, troubleshooting**

The history of Tokyo SRE team

Embedded SRE model

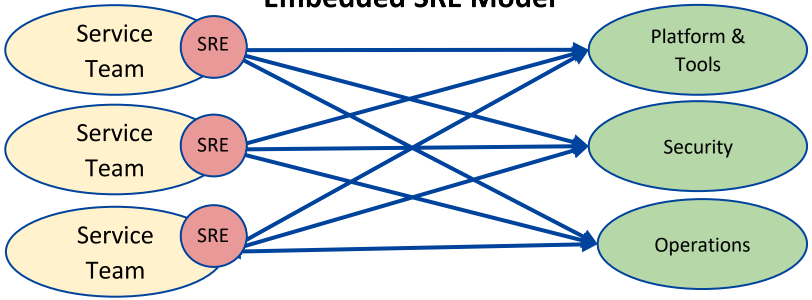
- 20+ SREs, 10+ scrum teams
- 2 SREs embedded in each scrum
- Reliability, Performance, Efficiency, Security



PS5 launch (2020~)

- Deployment, Monitoring, Alerting, Troubleshooting, Cost Mgmt as a scrum
- SRE collaboration across scrums when improving common operation scheme
- CI/CD and monitoring/logging tools provided by platform and tools team

Embedded SRE Model



Service Team

SRE

Service Team

SRE

Service Team

SRE

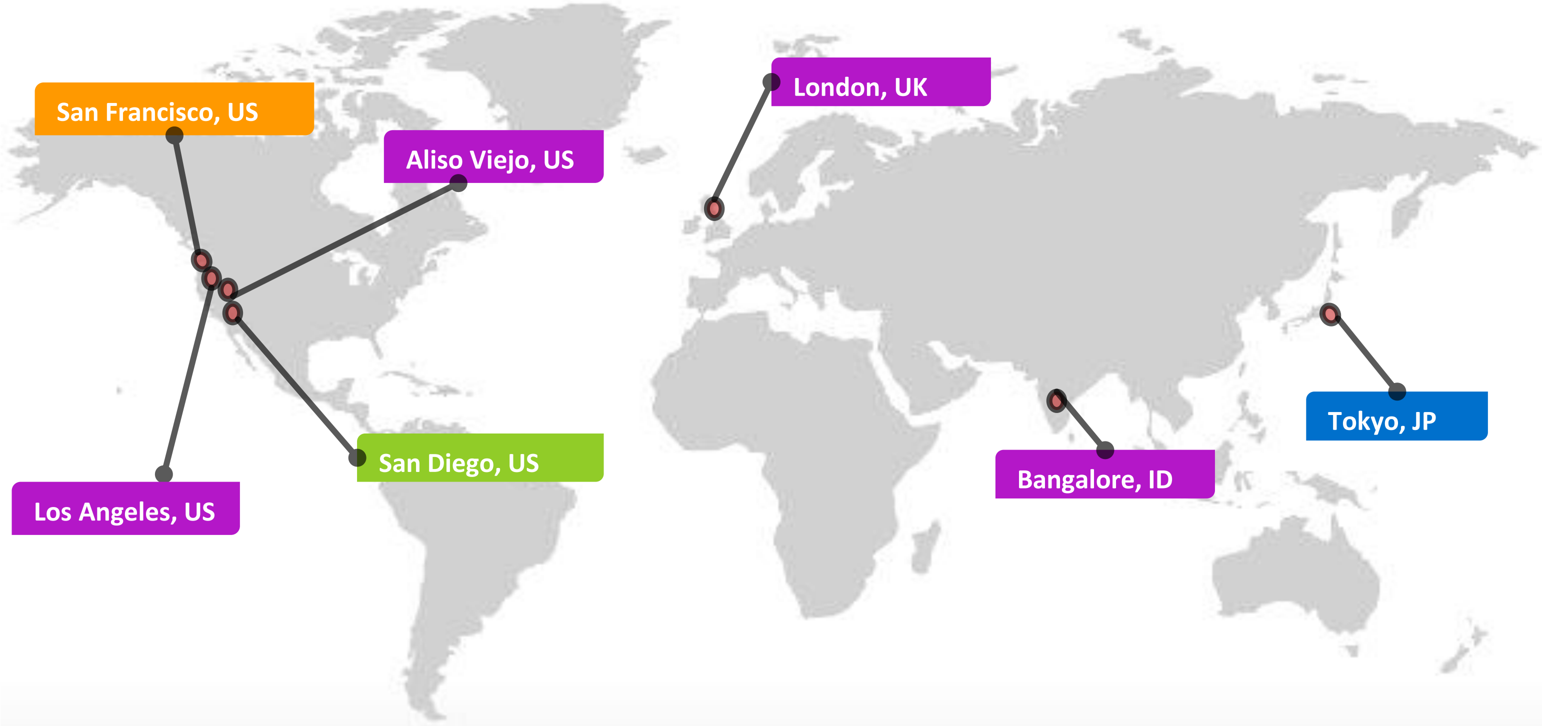
Platform & Tools

Security

Operations

Global SRE Overview in PlayStation Network

PlayStation Network Server team map



SRE Teams outline (focus on this session)

Tokyo	**Embedded SRE model** - 20+ SREs supporting 10+ scrum teams (200+ services)
San Diego	**Embedded SRE model** (Commerce) - 10+ SREs supporting 10+ scrum teams(50+ services) **Central SRE model** (Account) - 10+ SREs supporting 10+ scrum teams(50+ services)
San Francisco	**No SRE model** - Developers owning SRE/Ops tasks (200+services)

Other SRE Teams / Related teams

Other SRE Teams

- Los Angeles
- Aliso Viejo
- UK
- India
- Client, Studio, etc.

Related Teams

- DBRE Teams
- Infrastructure Teams
- Operation Teams
- Platform Teams

Embedded SRE model - Tokyo

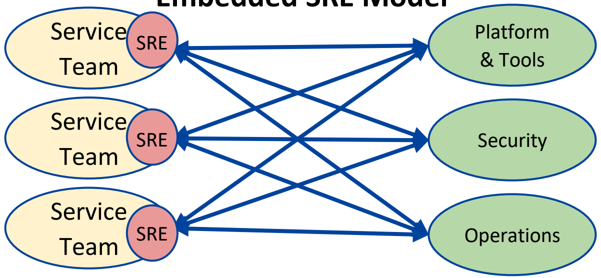
Going Well

- Stable SRE capacity per service
- Ease in maintaining service status awareness
- Mature DevOps culture

Opportunities

- Difficult to scale w/ the platform and organization
- Prioritization of Scrum Activities > Cross-Scrum Kaizen
- Sepecialized SREs – need to consider knowledge sharing + job rotations

Embedded SRE Model



Embedded SRE model(w/ Core SRE team) - San Diego(Commerce)

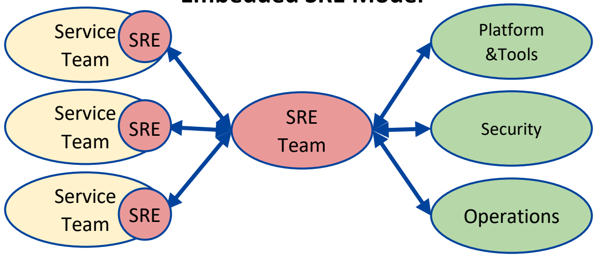
Going Well

- Service teams(Dev) can focus less on Ops
- Less dependencies
- Core SRE team facilitating cross-service collaboration

Opportunities

- Dedicated embedded SRE turns into THE “ops person”
- Difficult to scale w/ the platform and organization
- Dedicated embedded SRE turns into THE “ops person”

Embedded SRE Model



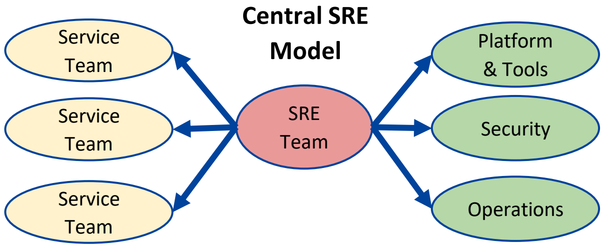
Central SRE model - San Diego(Account)

Going Well

- Centralized backlog and priorities
- Easier to standardize on process, tools and adoption
- “One stop” to troubleshoot/resolve issues across services
- Provide single interface for external stakeholders

Opportunities

- Scaling team and align priorities among dev, SRE, and other teams
- Coordination between service and platform/security teams
- Service team push toil work on SRE



No SRE/ad-hoc SRE model - San Francisco

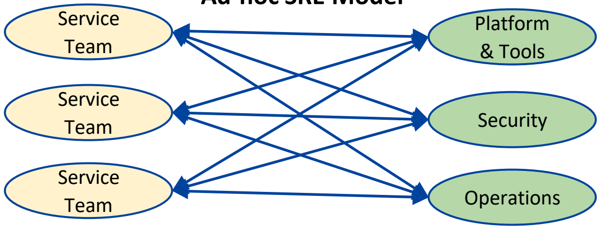
Going Well

- Empowers service teams: “you build it, you run it”
- Ensures service teams understand technologies and dependencies in-depth

Opportunities

- Less mature service teams still reach out to k8s platform team
  - *“my service is down, can you help me figure it out?”*
- Missing SRE to help improve the service for the future

Ad-hoc SRE Model



Model	Pros	Cons
Embedded SRE	- Stable SRE task progress - Focus on each scrum's services - Utilize Dev&Ops expertise in one team - Mature DevOps culture/team autonomy	- Lack of centralized SRE backlog - Headcount scalability - Multiple interfaces for external teams - Tend to cause personalization of tasks
Central SRE	- Stable SRE task progress - Centralized backlog - Single interface for external teams - Easy to share knowledge among SREs	- Boundary between Dev and Ops - Cognitive load for services' variety - Coordination difficulty
No(ad-hoc) SRE	- Focus on each scrum's services - Maximize team autonomy - Grow in-depth system understanding	- Unstable SRE task progress - Lack of centralized SRE backlog - Multiple interfaces for external teams - Require team maturity

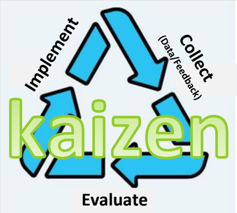
Key Takeaway

- No single SRE model fits for all teams
- Each SRE model can solve different problems
- The function of the model is influenced by it's environment:
  - Culture, System, Member capability, Team maturity

SRE leaders – Simply selecting a model is not enough

To fully optimize we must intimately understand and consider several factors

- Your Team's Strengths
- Your System + Evironment
- Observation Metrics
- Your Team's Input + Voice
- Four Keys + Availability



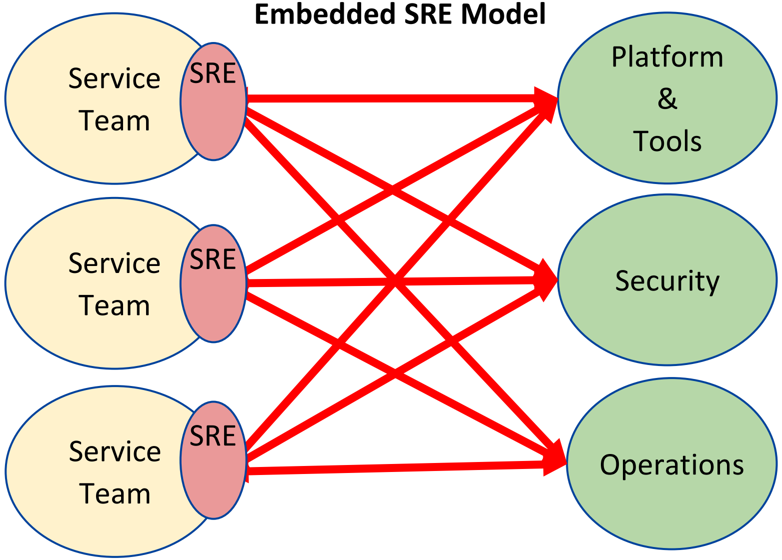
Let's use kaizen to polish the SRE team together!

Tokyo SRE team for the future

Opportunities for Embedded SRE model

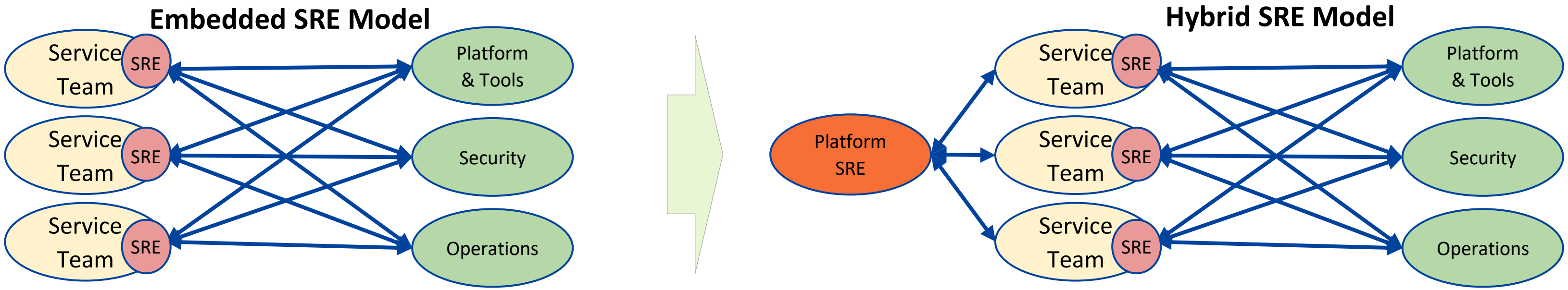
- Difficult to scale w/ the platform and organization
  - **Embedded SRE needs to do platform enhancement work and customer delivery work in parallel**
  - **Unifying infrastructure/tools and security enhancement are more important as the platform become bigger**
- Prioritization of Scrum Activities > Cross-Scrum Kaizen
  - **SRE needs to do scrum work and cross-scrum kaizen work in parallel**
  - **Difficult to prioritize cross-scrum kaizen work since SREs are embedded in a scrum**

Embedded SRE Model



Hybrid (Embedded and Platform) SRE

- Launch Platform SRE team this year, and migrate our SRE model to Hybrid SRE
  - **Objective 1**: Improve efficiency of cross-scrum service operations work
    - Standardize operation baseline so that SRE can apply changes across the services
  - **Objective 2**: Reduce the Embedded SRE workload, allowing increased focus on “SRE work”
    - Leverage the resources and assets that other related teams have
  - **Objective 3**: Introduce new model and approach improvements to embedded SREs + scrums



HORIZON
FORBIDDEN WEST™
Captured on PS5™