Заключение

Иван Круглов
SRE

slurm.io

SLO SLI SLA

SLO SLI SLA

- **SLI – что измеряем**

Тип	Измерение	Описание
API	availability	соотношение запросов, на которые был успешно получен ответ, и общего количества ответов
API	latency	соотношение запросов, на которые ответ был получен быстрее определенного порога, и общего количества ответов
API	quality	соотношение запросов, на которые ответ был получен с пониженным качеством, и общего количества ответов
storage	durability	соотношение сохраненных записей, которые могут быть успешно прочитаны, и общего количества записей
pipeline	freshness	соотношение данных, которые были обновлены раньше определенной точки во времени, и общего количества данных
pipeline	correctness	соотношение записей, обработка которых выдала корректный результат, и общего количества записей
pipeline	coverage	для групповой обработки - соотношение обработанных записей для потоковой обработки - соотношение записей обработанных за определенный временной интервал

source: Site Reliability Workbook

SLO SLI SLA

- **SLI – что измеряем**

- SLI – что измеряем
- SLO – какое целевое значение у того, что измеряем

СЛЁРМ

- SLI – что измеряем
- SLO – какое целевое значение у того, что измеряем
- Error Budget = 100% – SLO
- Error Budget Policy – алгоритм действий, если потратили бюджет

SLO SLI SLA

SLA – Service Level Agreement

SLA – контракт:

- предоставляемый сервис
- его характеристики
- обязанности и ответственности сторон
- последствия нарушения договоренностей
- …

на практике:

- внутри компании часто используют только SLO
- SLA работает между заказчиком/подрядчиком

- SLI – что измеряем
- SLO – какое целевое значение у того, что измеряем
- Error Budget = 100% – SLO
- Error Budget Policy – алгоритм действий, если потратили бюджет

- SLI – что измеряем
- SLO – какое целевое значение у того, что измеряем
- Error Budget = 100% – SLO
- Error Budget Policy – алгоритм действий, если потратили бюджет
- SLA – документ, суммирующий и юридически закрепляющий договоренности

Ценные Указания

СЛЁРМ

С чего начать?

- не переживайте, если с первого раза не получится!
- простейшие примеры SLI и примерные значения SLO
- собирайте статистику, считайте Error Budget
- отмечайте, когда и почему больше всего тратится бюджет
- приоритезируйте эти задачи – это Error Budget Policy
- регулярно пересматривайте SLO/SLI, отражайте User Experience

это бесконечный цикл :-)

Что мы изучили на практике?

Что мы изучили на практике?

- Работа в команде над incident response
  - Роли firefighter, incident leader, communication leader
- Практика observability
  - SLI/SLO для Availability, Latency, Freshness
  - Алерты
  - Бизнес-метрика
- Паттерны построения надежных систем:
  - Retry, Timeout, Request context
  - Health-checks
- Канареечный деплой через Argo Rollouts

SRE onboarding

орг. модель SRE

встроенный SRE

- SLI/SLO
- reliability practices
- canary releases
- oncall
- incident management
- blameless culture
- postmortems

централизованный SRE

- infrastructure
- tools
- disaster-recovery planning & testing
- SLI/SLO reviews
- central on-call teams

Начальный этап

- Несколько SRE энтузиастов
  - помощь определения и настройки SLO/SLI
  - reliability practices
  - on-call & incident management
  - postmortems
- Ключевые бизнес сервисы
- Некоторая поддержка руководства

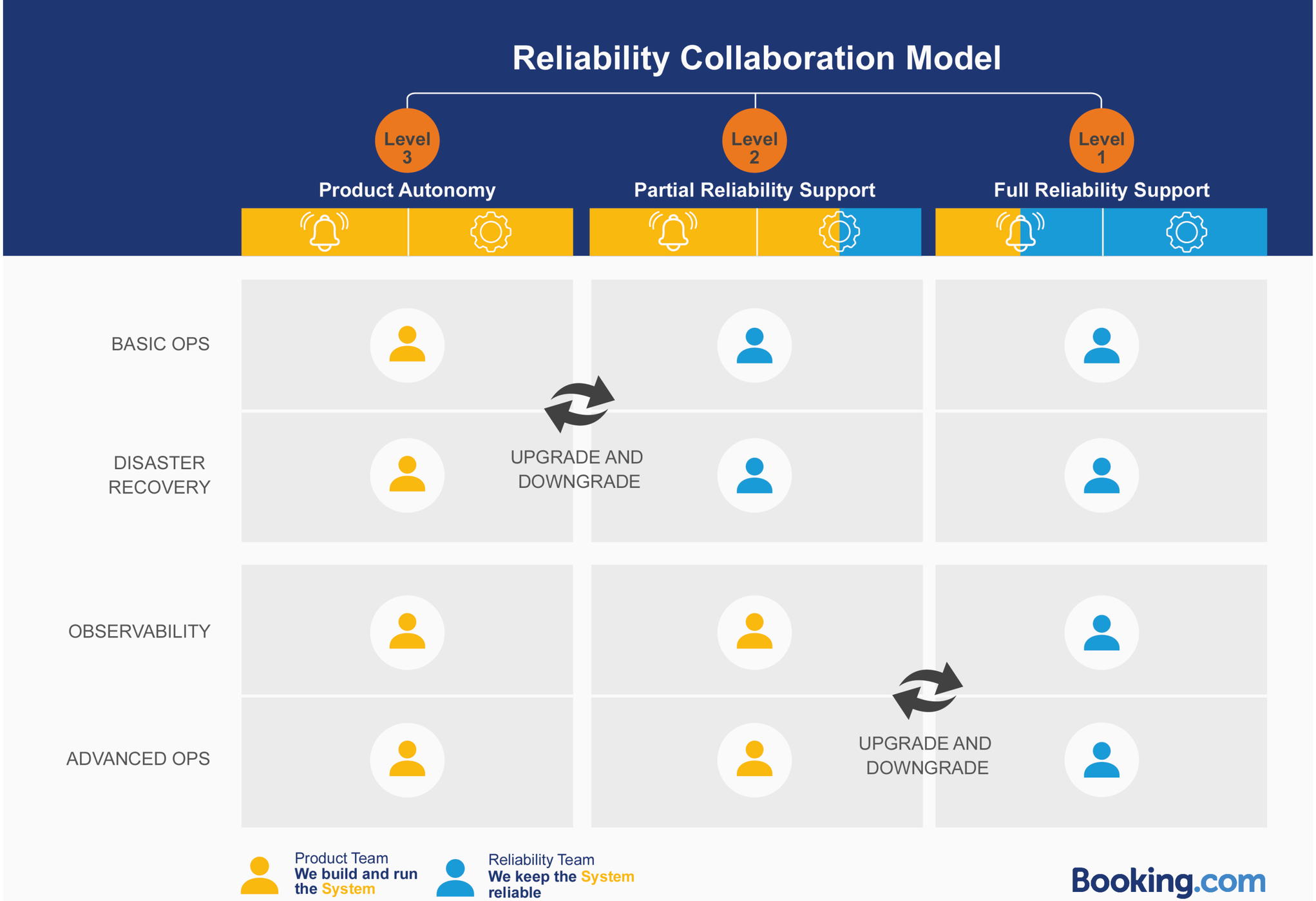
Следующий этап

- Центральная команда SRE
- Инфраструктура
  - canary release
  - service mesh
  - chaos-monkey
  - capacity planning
  - frameworks / templates
  - …
- SLI/SLO weekly review

Продвинутый этап

- Центральный on-call для зрелых сервисов
  - Критерии принятия сервиса в поддержку
- Disaster-recovery planning
- Drill days
- …

Reliability Collaboration Model



What is the Reliability Collaboration Model

The Reliability Collaboration Model, or RCM, is a model that solves two things. First, it maps out all reliability responsibilities for an infrastructure system, and second, it clarifies who among the reliability and product teams will own those responsibilities. The RCM is meant to be read from the point of view of a system.

A companion tool, called the *Ownership Map*, can be used by reliability teams to map out all the systems that they support.

Reliability Consultancy

The reliability teams provide consultancy to anyone and at any time, regardless of their level of support or if they are supported at all. This includes:

- FAQ and documentation
- Answers to ad-hoc questions
- Ad-hoc trainings on reliability topics
- Architectural reviews of systems at any stage of their development, including production readiness reviews before a system goes into production, and early stage reviews when a system is getting designed.
- Help to create bare-metal server roles.

Go to http://go/askreliability and get help now!

Incident Handling

In Level 3 and 2, the incident handling is owned by the product teams, and in Level 1, the ownership is shared between the product teams and the reliability team.

Owning the incident handling of a system means that the team decides what pager alerting they want to setup for the system, and what reaction time they want to give to the system in case of an outage. The expectations of quality of service and the SLOs from the customers have to be taken into account.

Reliability tasks

These are the activities required for a system to be healthy. Note that in Level 3, Disaster Recovery and Advanced Operations are the responsibility of the product teams, however systems in Level 3 often have low-enough criticality that those topics can be ignored.

BASIC OPS

All required for Level 2 and 1

- Service Level Indicator and Objectives (SLIs/SLOs)
- Adhering to compliance, security, and engineering guidelines
- Standard reporting of errors (Events and Rsyslog)
- Standard configuration management
- Standard deployment tools
- Server provisioning ops
- Server and package upgrades ops
- Capacity planning and stress testing
- Customer usage distribution mapping and reporting
- Dependency mapping

DISASTER RECOVERY

All required for Level 2 and 1

- Traffic load balancing
- Multi availability zone setup
- Datacenter failovers
- Automated data backups and recovery
- Regular testing of disaster recovery processes

OBSERVABILITY

All required for Level 1

- Alerting
- Monitoring
- Opdocs/Playbook
- Documentation following standard templates

ADVANCED OPS

All required for Level 1

- Dependency testing: chaos monkey and graceful degradation detection
- Traffic overload management: rate limiting, throttling, etc.

Upgrade and Downgrade

Systems can get upgraded, which means more of the reliability tasks are getting done by a dedicated reliability team, or they can get downgraded, which means the reliability tasks get done by one of more product teams.

Upgrades and downgrades can be requested by emailing the team leaders and project managers of the dedicated reliability teams for those systems.

LEVEL UPGRADE

Product teams must justify their request to upgrade a service based on its business criticality or amount of traffic. If the reliability team cannot take the additional work based on their *Ownership Map*, then the product organization will send extra engineers or downgrade other systems.

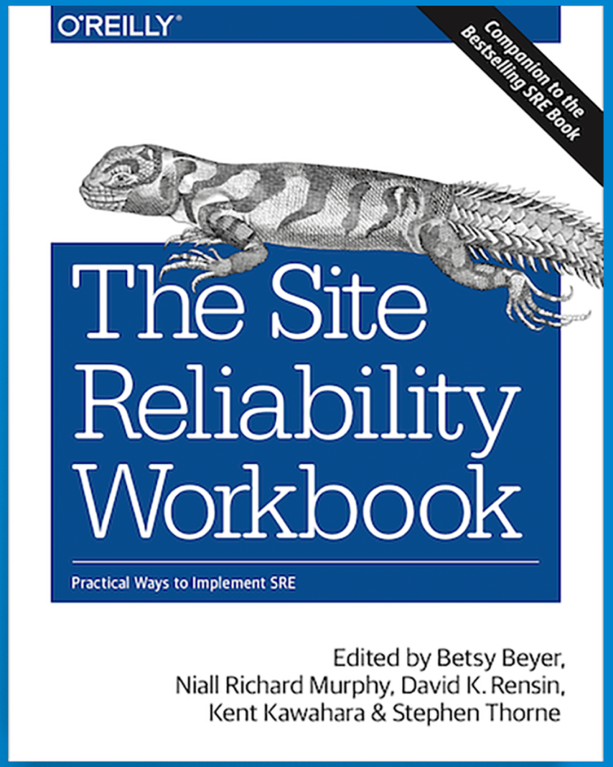
The system will need to be improved so it passes the quality requirements of the next level. That work will be split between the product and reliability team, which can be discussed during the initial conversations.

LEVEL DOWNGRADE

Reliability teams will initiate a downgrade if a system consistently violates its SLOs or sends false positive alerts due to product-related code, or if a system breaches the Security, Compliance, or Engineering guidelines. In that case, the product teams will also become full owners of the incident handling.

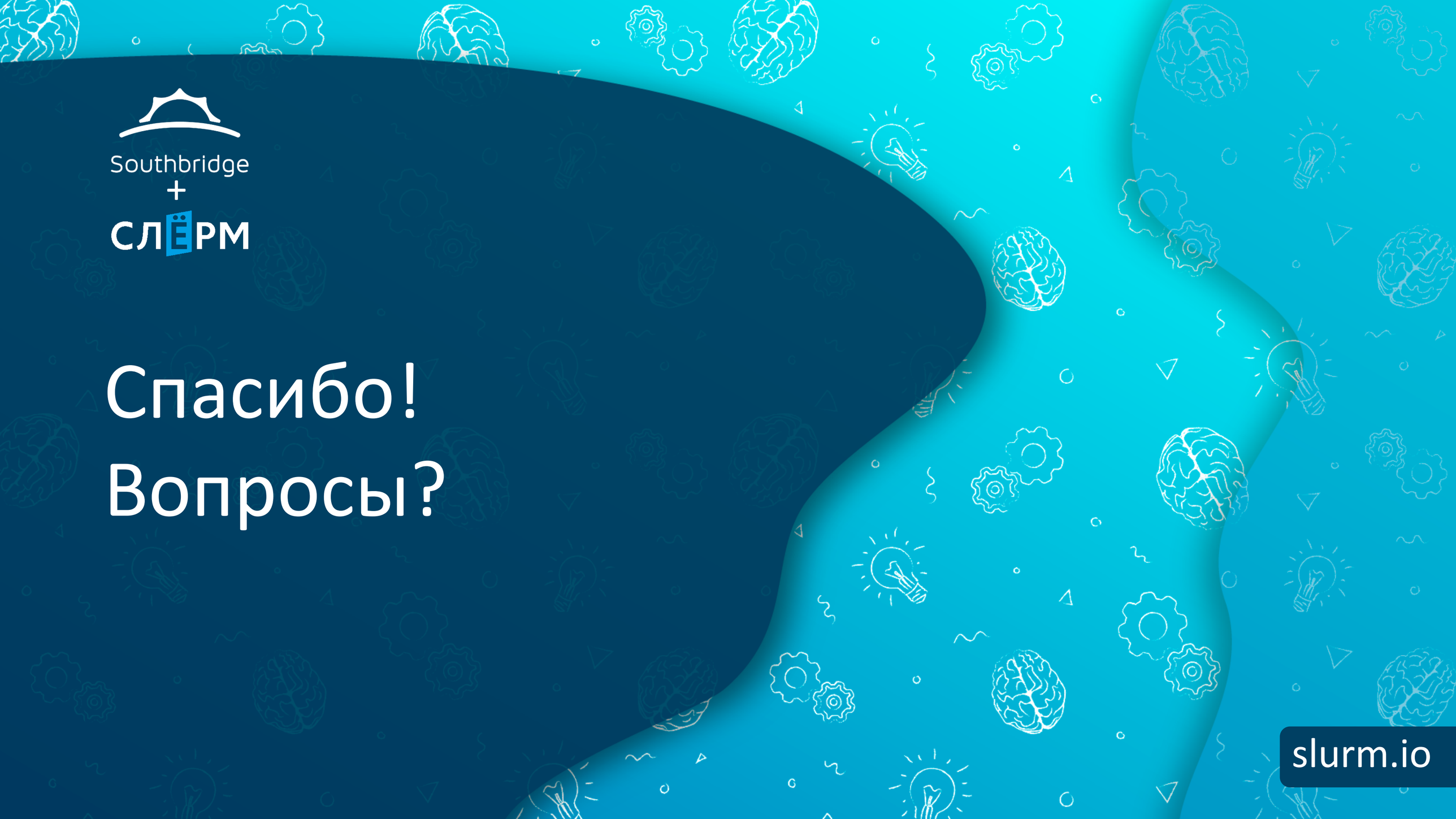
Booking.com

1. https://cloud.google.com/blog/products/devops-sre/how-sre-teams-are-organized-and-how-to-get-started
2. https://cloud.google.com/blog/products/devops-sre/how-to-start-and-assess-your-sre-journey
3. https://medium.com/booking-com-infrastructure/how-reliability-and-product-teams-collaborate-at-booking-com-f6c317cc0aeb



СЛЁРМ

Иван Круглов
mail@ikruglov.com
https://getmentor.dev/mentor/ivan-kruglov-16



Спасибо!
Вопросы?

slurm.io