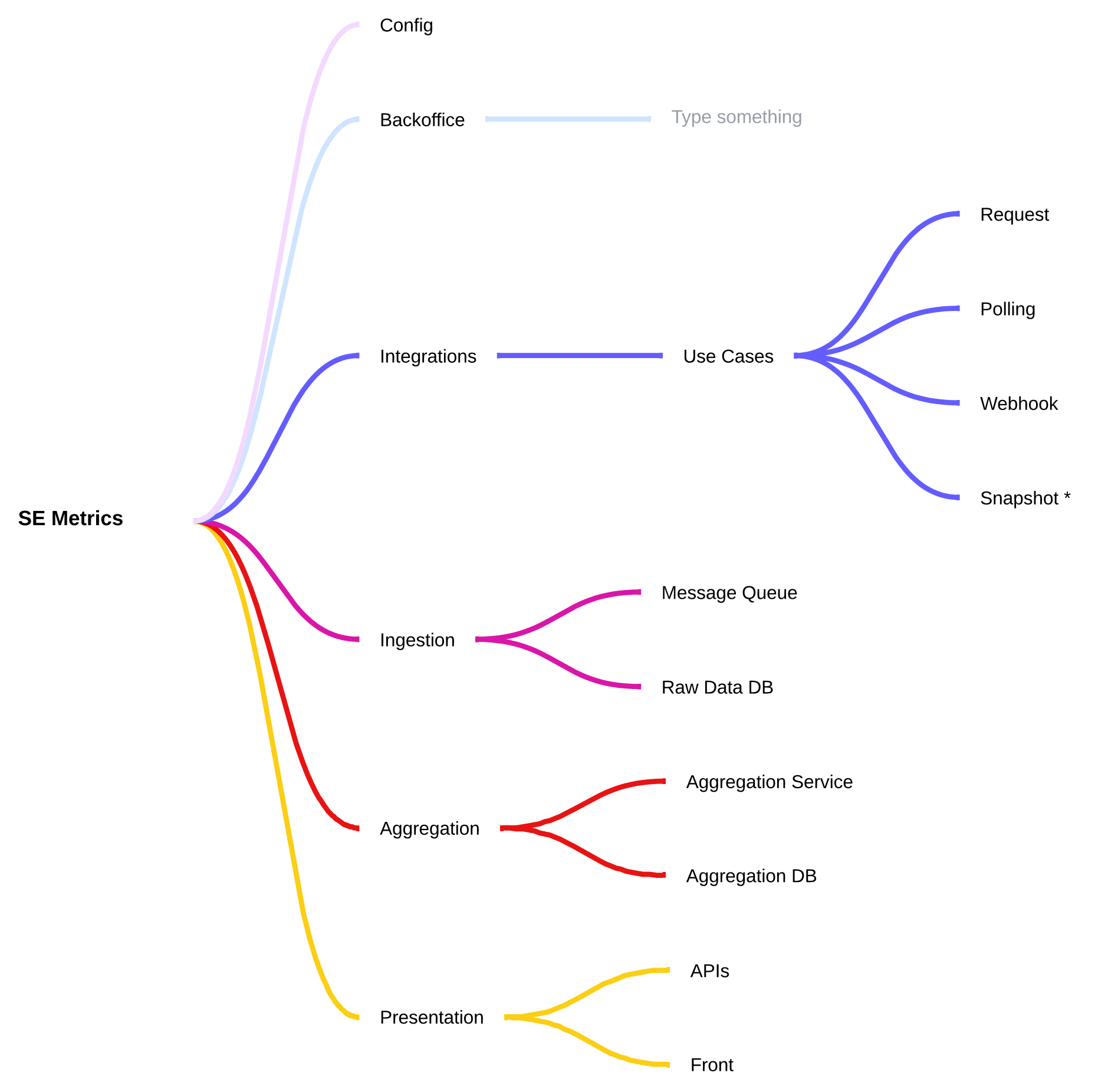

Simple sprint retrospective
Created by Leo Barnes
Leo Barnes

Team retro

1. Review the actions from the **last** retrospective (5 min).
2. Think about the current sprint. Add things that went well and things that need improvement (10 min).
3. Group the items and discuss.
4. Use the voting feature in "Team Tools" to vote on the items in the "Needs improvement" zone. The 2-3 most voted items should be actioned for the **next** sprint (10 min).

Tip: Use Lucidspark Cards and a Lucidspark timeline to keep track of action items moving forward.

Config & Onboarding

- DB com metadados de integração para configurações iniciais
- Variáveis de ambiente para tokens e api keys
- Evitar sobreposição de dados e workloads desnecessários (Aggregation Window, Data deduplication, Delivery garantees, etc.)

Snapshot

- Setup inicial que utiliza requests para busca de dados hitóricos
- Deve assinar os webhooks para atualização periódica dos dados
- Deve ser utilizado sempre que um novo tipo de dado passa a ser consumido
- Configuração de webhooks
- **Reconciliation** (apenas para lembrar de analisar se há necessidade)
- Meta-informações configurações

Backoffice

- Banco de dados relacional
- Criação, configuração e edição de usuários, contas, etc.
- Multi-tenancy
- Configuração de integrações
- Frontend
- API
- Backend

Collection

Como eu me comunico com os vendors

Request

- consulta não recorrente de dados históricos
- consulta não recorrente de dados pontuais
- Configuração de webhook
- O serviço é responsável pela padronização do payload que será enviado parao MQ
- Reativo; depende de algum evento para realizar uma requisição

Polling

- consulta recorrente de dados que não possuem webhook
- A frequencia do polling depende da natureza dos dados e pode ser variavel
- Gestão de Rate Limit
- O serviço é responsável pela padronização do payload que será enviado parao MQ

Webhook API

- Envio de dados pelos vendors para os endpoints cadastrados
- APIs responsáveis por receber dados dos vendors
- Uma API para cada categoria de dados e uma camada anticorrupção. Ex.: Issue Tracking, CI/CD, etc.
- O serviço é responsável pela padronização do payload que será enviado parao MQ

Ingestion

Como eu esturuturo os eventos e os workloads

Message Queue

- Rabbit MQ
- Cada tipo de dado será enviado para um endpoint
- Cada tipo de dado será ingerido em um tópico

Raw Data DB

- A princípio, os dados brutos serão enviados para um bucket S3

Aggregation

Como eu transformo e organizo os dados para otimizar a criação e exibição das métricas

Aggregation Service

- Inicialmente seguirá o fluxo síncrono inserindo direto no DB
- Deve ser modularizado para facilitar a separação em microsserviços no futuro
- A separação em ms deve ser inicialmente por vendor e no futuro se especializar para cada tipo de entidade

Aggregation DB

- Os schemas serão definidos posteriormente a medida em que os payloads forem analisados
- entender melhor os tipos de visualizações para criar schemas mais aderentes
- Banco de dados de documentos, inicialmente documentDB

Presentation

Como eu apresento as métricas

API

- API para consumo de dados para construção de dashboards
- Consumo de banco de dados de documentos com dados agregados
- Consumo de dados indexados no elastic search

Frontend

- Dashboard e métricas
- Visão gestão
- Visão desenvolvedor

Leo Barnes