




FEATURE STORE SUMMIT

12-13 OCTOBER | 08:30 AM - 4:00 PM PT

ORGANIZED BY HOPSWORKS



Feature Store for Multi-Tenant and Multi-App in Salesforce



Alex Araujo
Principal Engineer
Salesforce



Weiping Peng
Principal Architect
Salesforce



Our Multi-tenant Architecture

Data at Salesforce is owned by our customers

- Strict approval from customer to use data
- Strict isolation of data

Customizable data model per tenant

- Each customer's data model is unique

Our Scale



Multi-Tenant

- Over 150K customer orgs
- Over 10B transactions/day
- Extendable data models



Multi-Apps

- Over 50 ML/AI GA applications
- App Template offering



Substrate Agnostic

- Customer's choice



Add Feature Store to ML Platform



Common Benefits

Enable collaboration

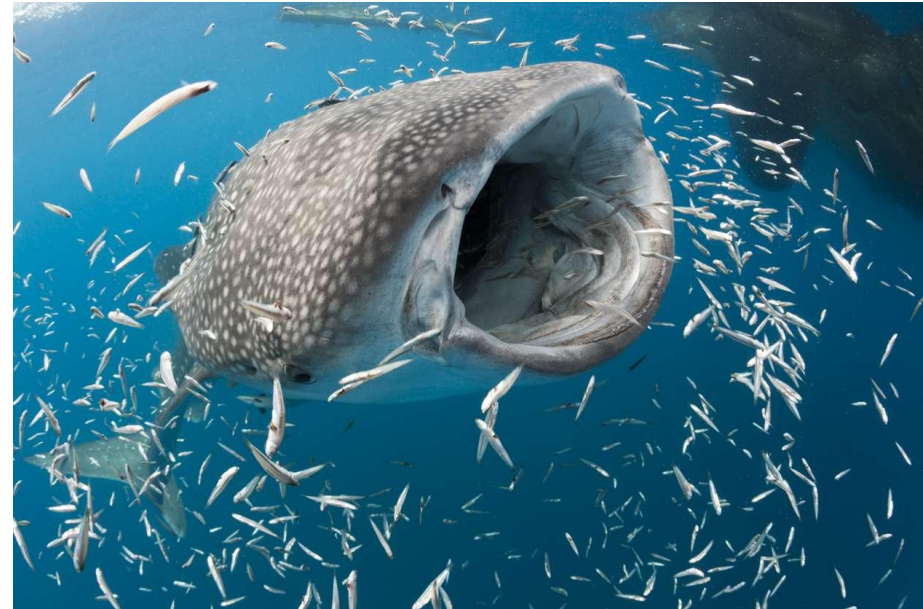
- Among data engineers, ML engineers, and data scientists

Operationalize feature generation

- Orchestrate and automate
- Metadata, data and feature lineage
- Secure, observable and traceable

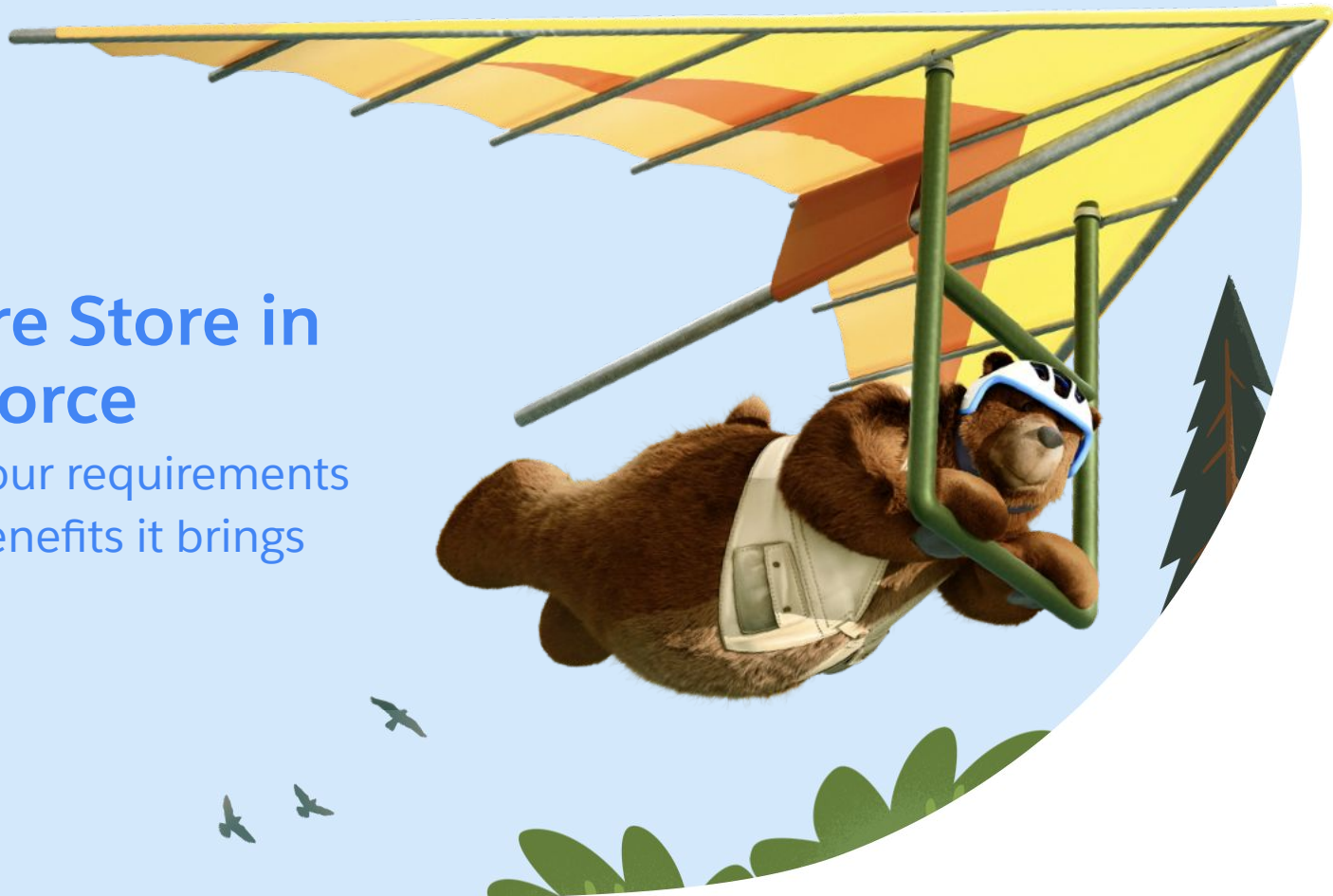
Training and Serving time consistency

- Compute once and reuse



Feature Store in Salesforce

- What's our requirements
- What benefits it brings



Access Model

Feature data requires the same protection as raw data

- Permission to generate and access is an extension of data access

Sharing rules

- Sharing feature data
- Sharing feature generation code

GDPR

- Applies to feature values too!



Metadata Driven

Use Salesforce metadata to drive feature engineering pipelines

○

Bake Salesforce metadata definition in feature lineage definition

Salesforce Metadata API:

https://developer.salesforce.com/docs/atlas.en-us.api_meta.meta/api_meta/meta_intro.htm



Evolve with Applications

Each app is different

Batch <> Streaming <> ondemand

- Feature computation and feature retrieval
- Keep interface consistent

Versioning

- Algorithm changes
-



Data Migration

Customer moves

- Data isolation
- Data and feature values move with customer



Enable Collaboration

Broader than Salesforce

Enable Data Owners to enrich their data

- Our external customers

Cross application sharing

- One CRM ecosystem
- One set of CRM data

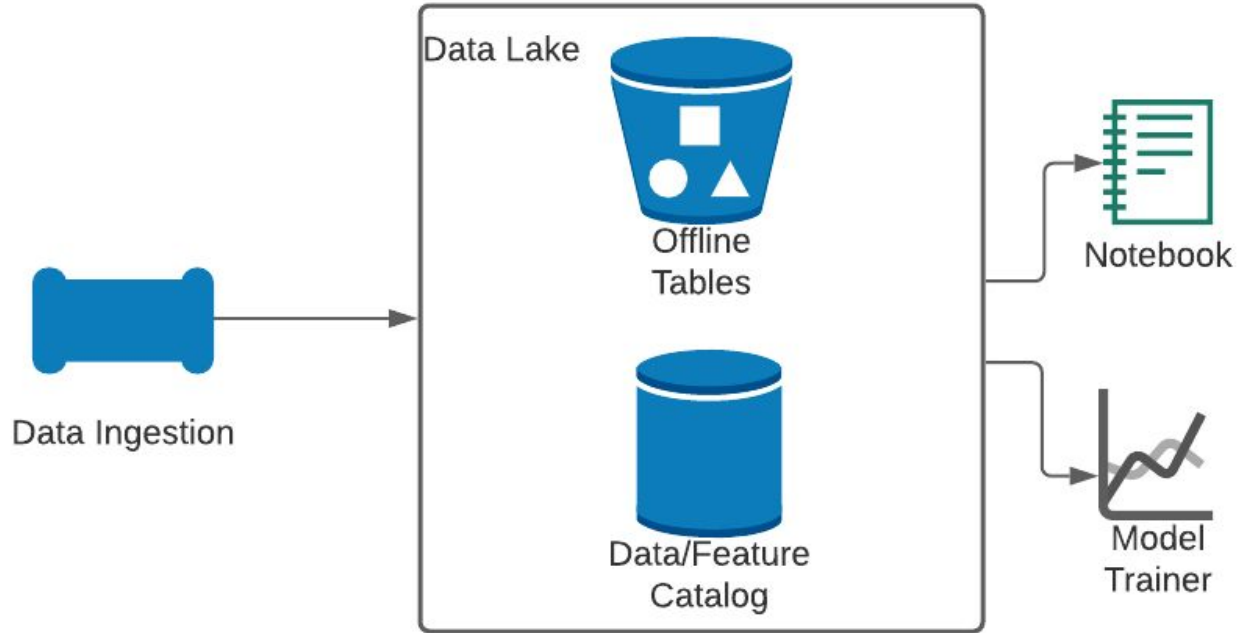


Architecture



Architecture Overview

System before Feature Store



Data Lake to Feature Store



Pipeline Deduplication

- Data Lake: ETL, feature generation, and model training for each model
- Feature Store: Single pipeline for feature generation can be used for multiple models

Batch Online Hydration

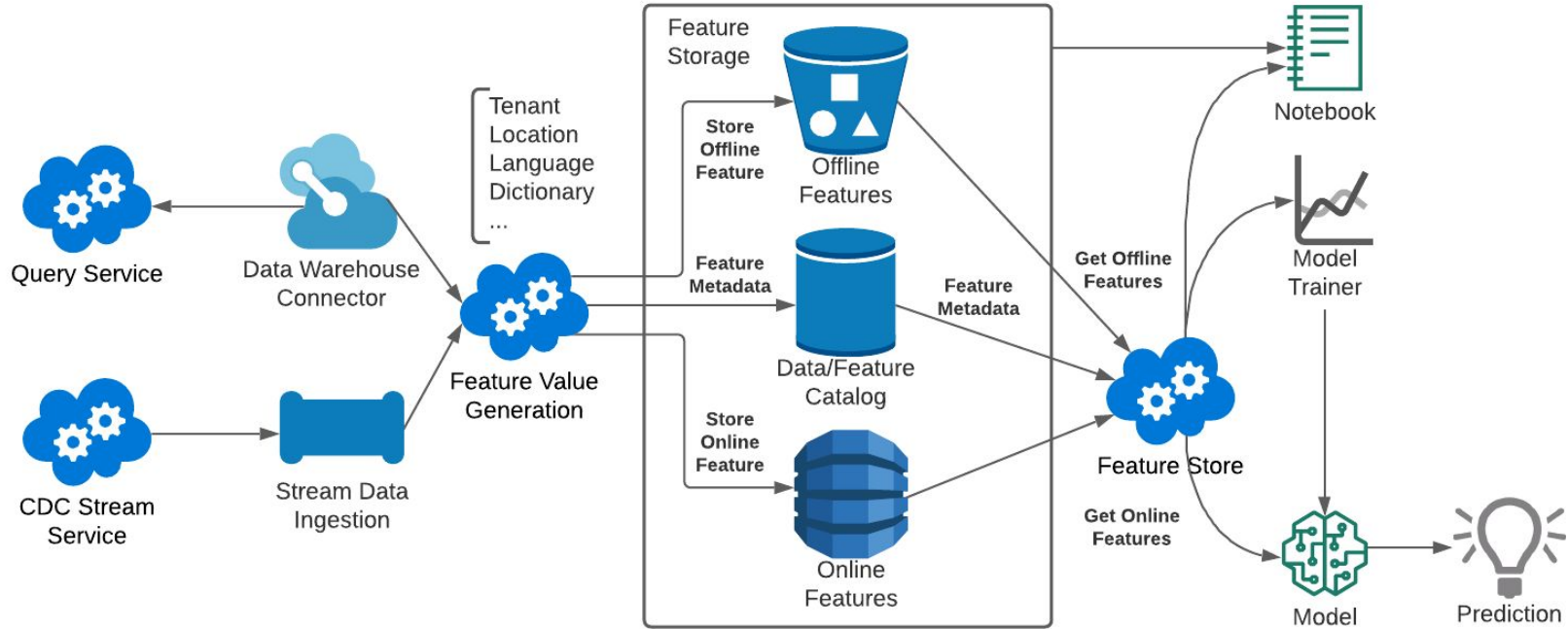
- Day 0: Create online feature tables from offline tables and copy features
- Incremental: Copy new features and updates

Extending Compliance

- Retention policies for feature tables derived from raw entity tables
- Feature tables stored in data catalog
- Tenant deletion jobs can find all tables and delete them



Architecture with Feature Store



Why Feast?



Feast is Open Source

- Compliments the rest of our stack
- Community is active and responsive
- Functionality and code organized and well documented

Feast is Extensible and Pluggable

- Easy to integrate with the rest of our stack

Feast's APIs/SDKs are simple and easy to use

- "We need something like Feast" -- Internal Users
- Desired language support included (Python/Java/Scala)



Challenges with Feast



Architecture and component instability

- Lots of changes between releases for Java components
- Java runtime no longer actively developed
- Python runtime does not have server components and diverging from Java runtime

Hard fork vs light fork

- Light forks based on stable upstream forks not feasible
- Enhancements to Feast remain on internal forks for a very long time

Compatibility across releases and runtimes would help

- API, metadata, and data compatibility contracts
- Python and Java runtimes adhere to the same contracts



Future Roadmap



Streaming

- Provisioning on-demand CDC streams
- Feature transformations for batch and streaming

Multiple Feature Versions

- Create new versions of features and model
- Generate previous and new versions of feature values
- Transition inference from previous to new feature values

Feature Sharing

- Deriving default sharing rules from customer entity permissions
- Allowing customers to view their feature catalog and customize permissions





Thank you!

Do you have any questions?

Speaker Info



<https://www.linkedin.com/in/araujoalex>

<https://www.linkedin.com/in/weipingpeng>



Thank You

