Personalization in multilevel e-commerce system

Lukas Grolig

Personalization

Is about providing customer his own user experience. Display features that he might need and promote products might buy.

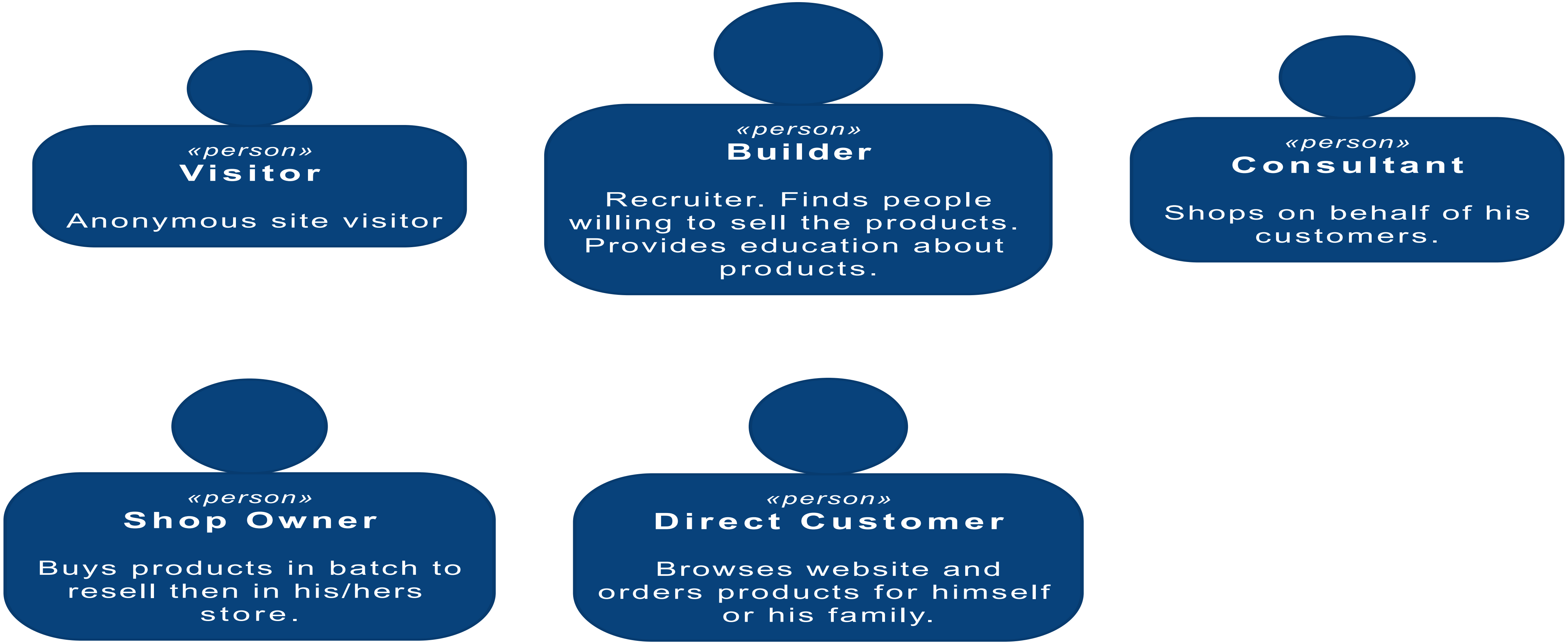
Multi-level marketing

Companies having multiple sales streams. Company is recruiting consultants - self employed individuals offering products. In multi-level there are also people responsible for recruitment of consultants as their main income. But company can also be selling directly to the customers.

Problems

When designing personalization in multi-level there are many types of users with completely different needs. Whole website can change so drastically that it is not about sales anymore.

Target persons



«person»
Visitor

Anonymous site visitor

«person»
Builder

Recruiter. Finds people willing to sell the products. Provides education about products.

«person»
Consultant

Shops on behalf of his customers.

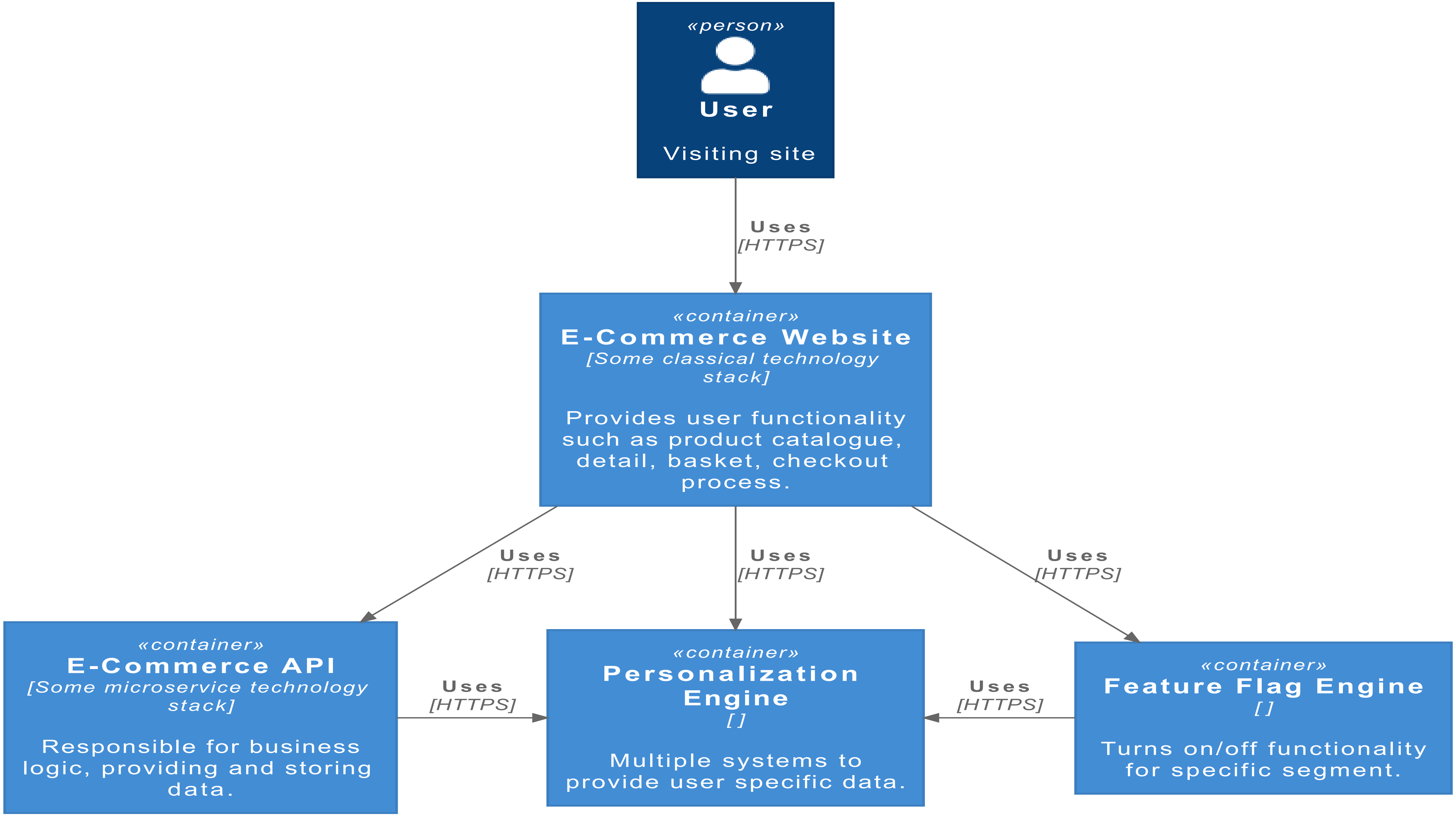
«person»
Shop Owner

Buys products in batch to resell then in his/hers store.

«person»
Direct Customer

Browses website and orders products for himself or his family.

System Design (Context)



«person»

User

Visiting site

Uses
[HTTPS]

«container»

E-Commerce Website

[Some classical technology stack]

Provides user functionality such as product catalogue, detail, basket, checkout process.

Uses
[HTTPS]

Uses
[HTTPS]

Uses
[HTTPS]

«container»

E-Commerce API

[Some microservice technology stack]

Responsible for business logic, providing and storing data.

Uses
[HTTPS]

«container»

Personalization Engine

[ ]

Multiple systems to provide user specific data.

Uses
[HTTPS]

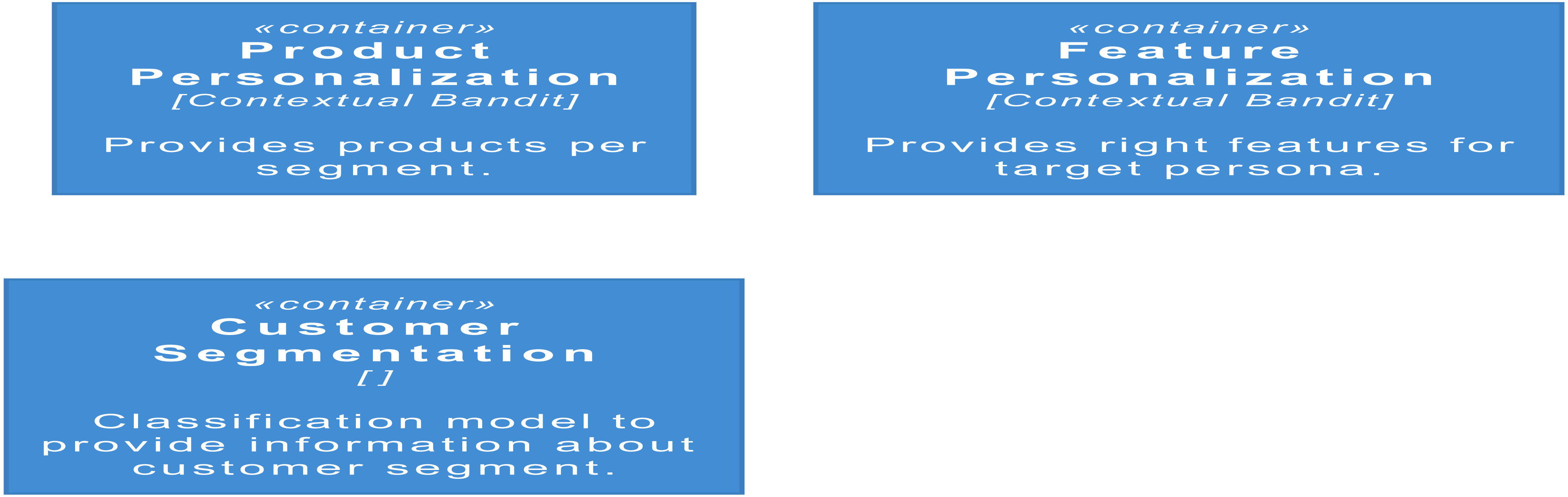
«container»

Feature Flag Engine

[ ]

Turns on/off functionality for specific segment.

Personalization Engine (Container)



«container»
Product Personalization
[Contextual Bandit]

Provides products per segment.

«container»
Feature Personalization
[Contextual Bandit]

Provides right features for target persona.

«container»
Customer Segmentation
[ ]

Classification model to provide information about customer segment.

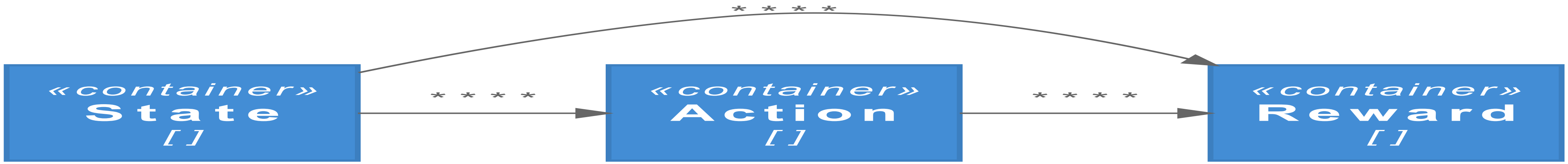
Segmentation

- offline experimentational training
- number of target groups selected by data scientist
- manual analysis of each segment (based on sales, features used)
- preparation for classification engine (defining features)

Classification

- part of ML pipeline
- based on segmentation analysis output, dataset is created after closings
- classification is based on Catboost (gradient boosting)
- models are trained, evaluated and deployed automatically

Contextual Bandit



Modeling the Problem

Context: product category, season, region

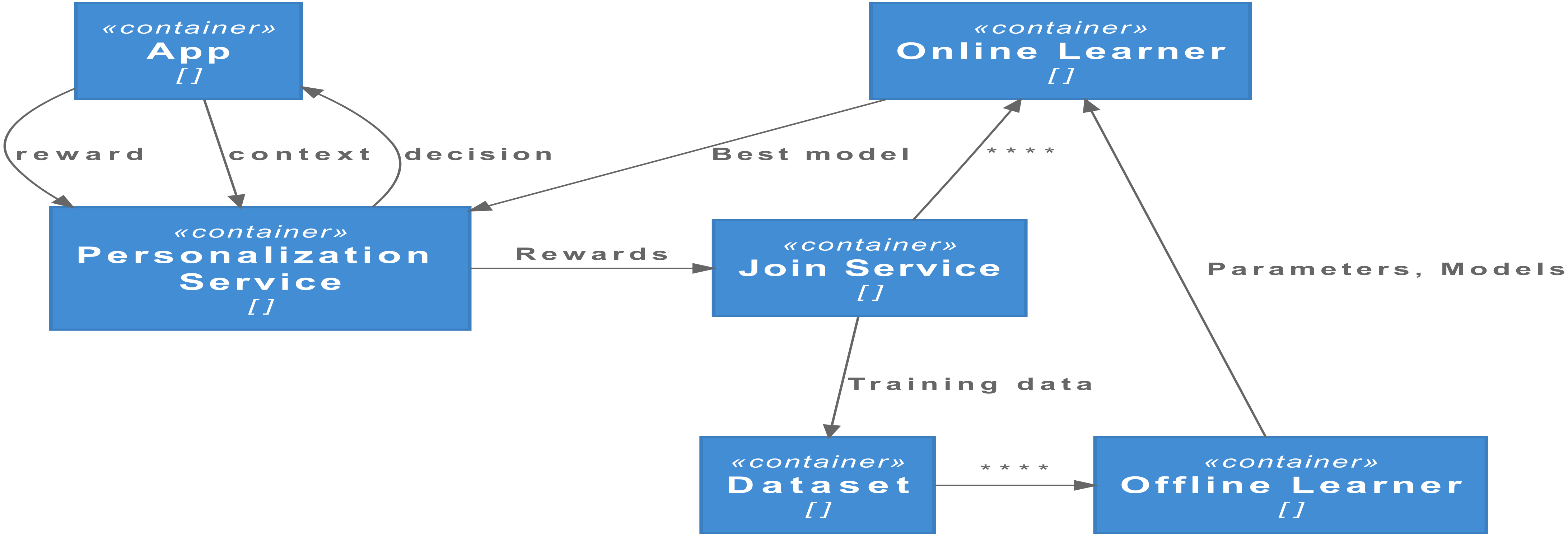
Decision: products to display

Feasible actions: { category1, category2, bestsellers, trending }

Outcome: user adds product to basket or displays product detail page withing 20s

Reward: 3 when added to basket, 1 when detail displayed, 0 otherwise

Contextual Bandit (Context)



Service Design

- each service must use telemetry to trace request through the system
- API to retrieve plain data is not accepting context
- personalization is separate API on top of plain

Personalized data source

- on top of data sources there can be personalizes data source accepting context

Problems:

- personalized sources are much slower
- caching can be challenging (memory consumption)

Personalized feature flags

- on top of regular feature flag service there is another providing features based on context
- enables complete customization of browsing experience
- affects frontend development and backend-for-frontend
- nearly every UI component has it's own flag
- components can be versioned (eg. suffix to feature folder)

Automated A/B Testing

- it is crucial to specify right metrics (display time, time to action, selling targets)
- each service must provide analytic outputs
- multiple metric sources must be aggregated (traces, google tags...)
- data are stored in a lake
- using ETL raw data are processed and stored in the lake again
- data are consumed by systems when needed

Join service

- 2 possible designs - offline based on ETL, online based on message hub (eg. Kafka)
- in microservice world it is the most complex part of the system
- heavy utilization of OpenTelemetry
- join service takes nearly all events same as logging service does

Questions?

Thanks a lot.