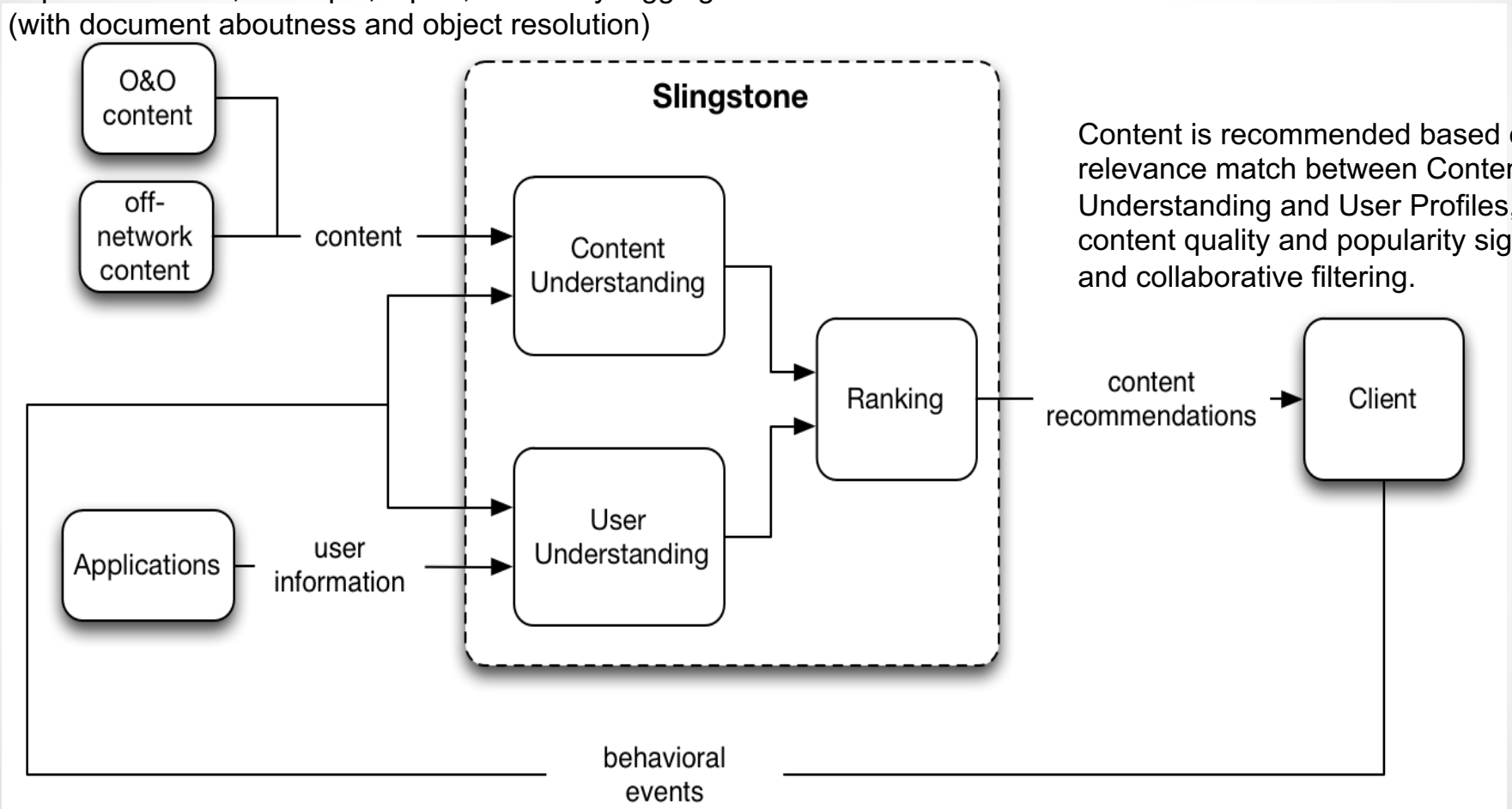
Introduction of Personalization Platform

How Personalization Works

How Personalization Works

Documents are processed by Content Understanding to produce terms, concepts, topics, and entity tagging (with document aboutness and object resolution)



Behavioral event data are processed by Content Understanding to produce features such as popularity, interestingness and by User Understanding to infer interests

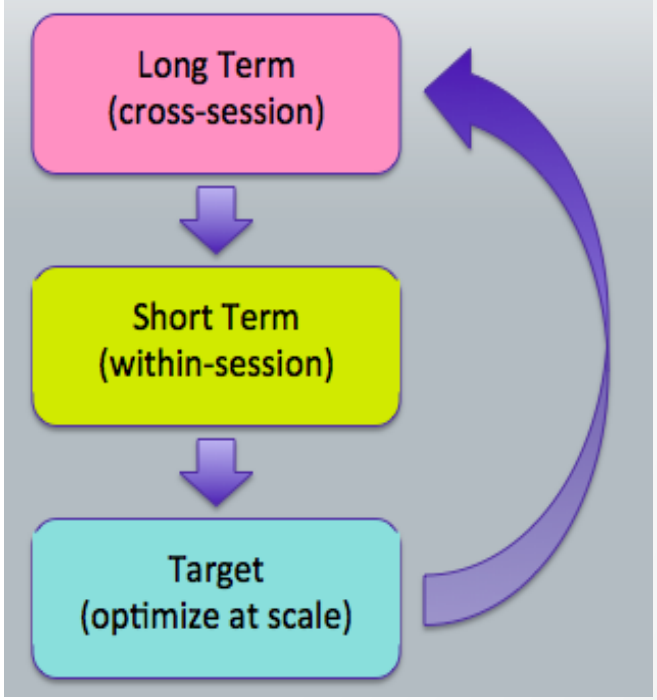
How We Measure Success

Defining long-term success

- **How many users we get**
  - Total number of stream-engaged users
  - Retention rate of new users
- **How often users visit**
  - Days visited per user per month
- **How long they stay**
  - Total stream dwell time per visit
- **How personalized is the experience**
  - User profile & dwell time lift
  - Percent of personalized visits

How we measure short-term success

Short Term Metrics
Relevance Dwell per Depth (DpD) Clicks per Depth (CpD)
Freshness Latency from publish to serving Latency from publish to ingestion
Comprehensiveness Competitive Content Coverage Topic Coverage
Speed Serving latency @ 99th and 50th p.
Cost Serving cost ($/DQ) Content cost ($/KDocs) Profile cost ($/KUsers)



Overview of How We Achieved Personalization

Overview: Content Understanding

Basic components

NLP

- Entity detection & resolution
- Aboutness (entity ranking)
- Categorization (topic, sensitive...)

Scalable understanding*

- n-grams
- Topics models (LDA)
- Story clustering / Dedup

Improvements

Authority*

- Web, Twitter, Facebook, etc.
- Domain, source, item
- NewsRank, TopicRank

Quality*

- Low/High quality, Controversial
- Newsy, evergreen
- Performance priors (features)

*Important for large content pools

Overview: Ranking and Recommendation

Basic components

Machine-learned Ranking

- Linear & GBDT
- Target & feature additions
- Online sparse logistic regression

Phase 1 Optimizations & Experiments

- Exponential age-decay
- Retrieval set size & diversity
- Improved WAND efficiency
- Negative-interest filtering

Infrastructure

- Search Platform
- Near-Realtime Pipeline (GMP/segmented, entity pop)

Improvements

Federated Retrieval & Ranking

- Parallel match types
- Personalized blending
- Response prediction model

Ranking Optimization

- Authority replacement for GMP
- WAND optimization
- YST/other page-features
- CF ranking
- Various match types

Unified Today Module & Stream

- Stream plus one w/ editorial signals

Overview: User Understanding

Basic components

Improved User / Interest Coverage

- SID profiles
- Large profiles
- Leverage off-net and all devices
- Feature tuning

Large-scale Experimentation System

- Test multiple profiles in parallel
- Unified feed of user events
- Build multi-segment profiles (property, device, etc.)

Profile Experiments

- TF variants, negative interests, source affinity, stream actions, and cold-start

Improvements

New Profile Models

- Contextual profiling
- Dynamic time decay

New Profile Signals

- Search, Mail, Social
- Exploring YDN, Flickr, Apps

User Interest Exploration

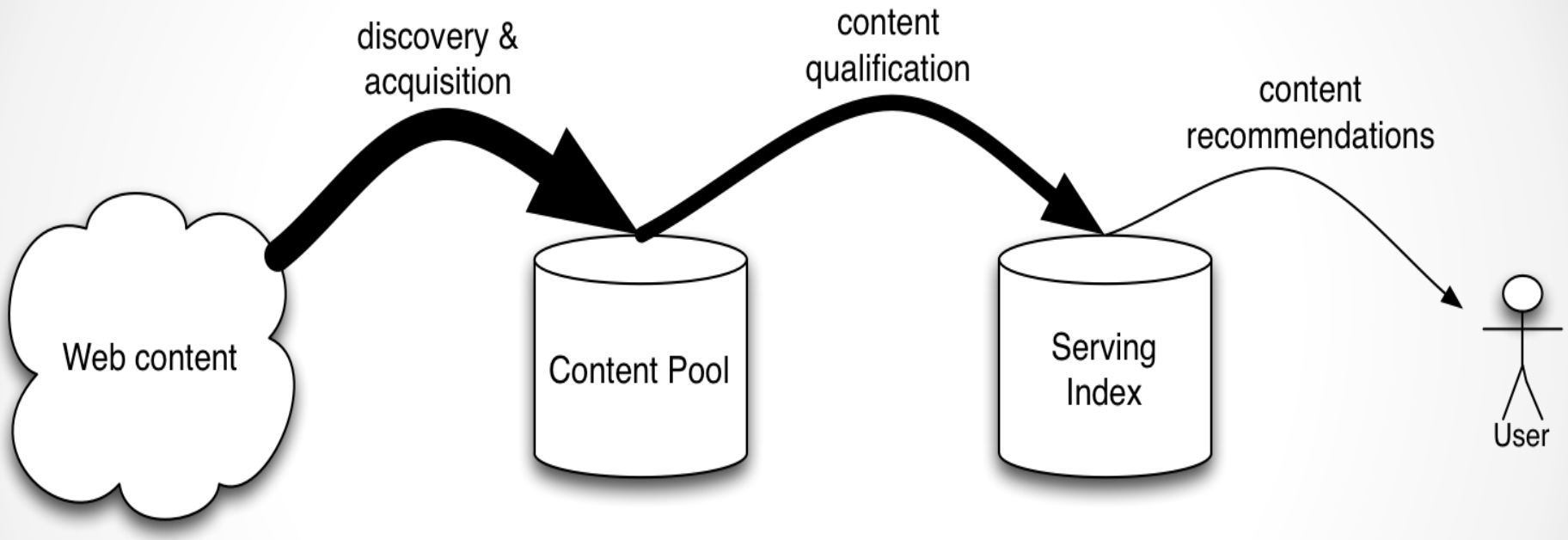
- Collaborative filtering: User-content factor model

User Segmentation Study

- Increase user retention
- Profile aspects importance

Expanding the Content Pool

Approach to Scale Content Acquisition



Rapidly scale content acquisition into Content Pool

Iteratively grow size of Serving Index while controlling for quality

Content Data Model

Context (per discovery type)

Metadata from discovery (e.g. RSS feed, Tweet time)

Content

HTML, HTTP headers

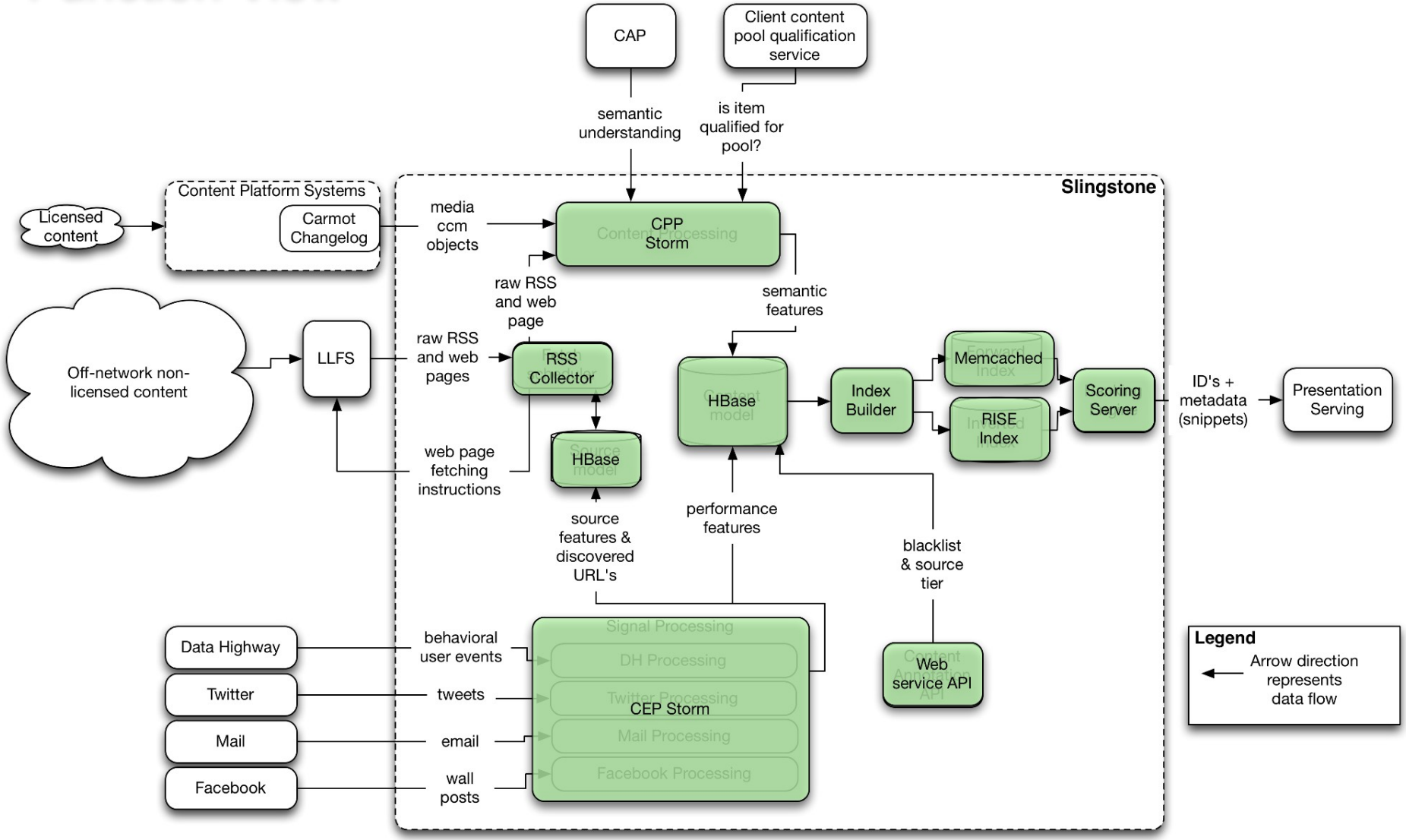
Metadata

Content-based features (e.g. classifiers: authority, quality, categories)

Signals

Behavioral features (GMP, Twitter score, Mail links, etc)

Function View



Approach to Grow the Serving Index

Offline Quality Scoring

Compute scores for every article in Content Pool using

- Classifiers: Adult, Junk, Low Quality e.g. Job Listing, PGA Tee Times
- Domain Quality & Authority Model

Only add to Serving Index docs that are above quality threshold

Large Online Serving Index

Estimate ~20M recommendable docs out of 100M content pool

Iterative approach to add docs while controlling for quality

Key technical challenges to solve

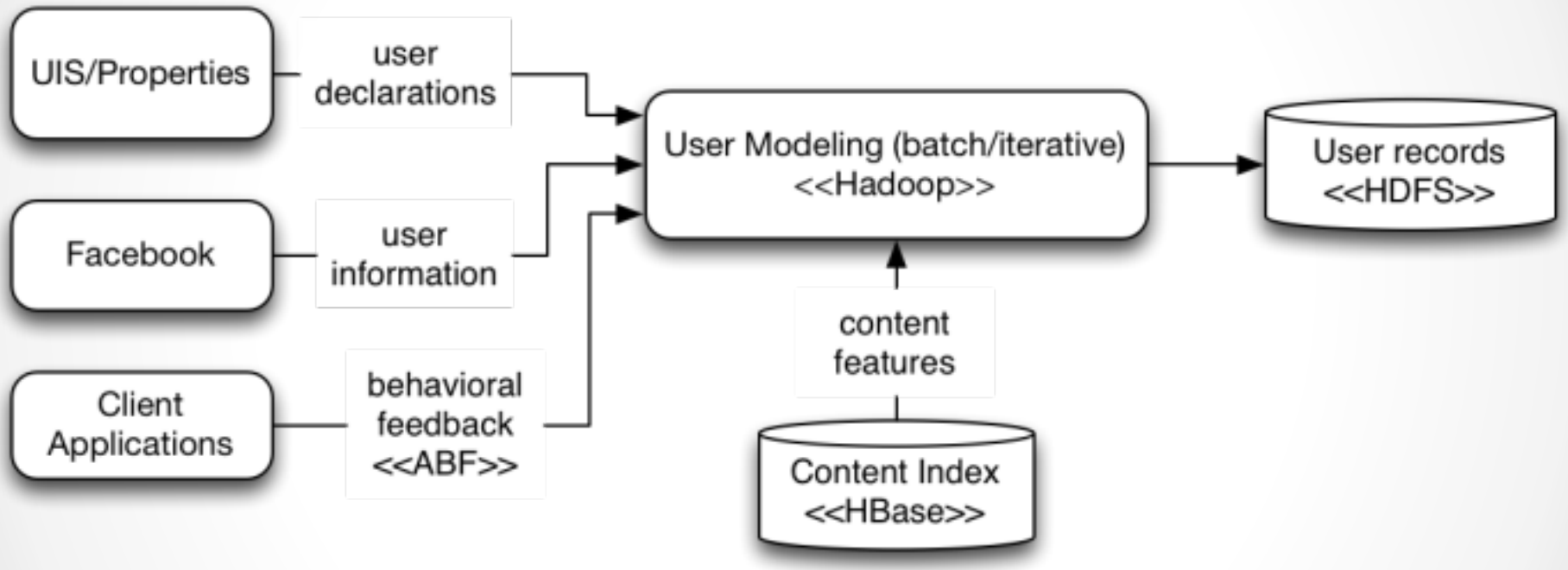
- Authority & Quality scores in Ranking
- Federated Retrieval and Ranking (scalable multi-match retrieval)

User Profiling

User Understanding

- Input Information
  - Declared user interests
  - Property application preferences for each user
  - Implicit behavior feedback
  - Social account information
- System Processing
  - Batch profile updates
  - Multiple simultaneous user models
  - User identity management
- Modeling
  - Positive inferred interests
  - TF-IDF
  - Sparse Polarity
  - Collaborative Filtering
  - Negative inferred
  - Contextual profiles

Function View



Ranking: From Relevance to Federated Ranking

Relevance Ranking

- Query
  - Query plans
  - Query expansion
- Document
  - Partitioned and replicated index
  - Near real-time index updates
- Ranking model
  - Phase-0: Weighted set/WAND for candidate retrieval
  - Phase-1: linear function with popularity, contextual, and personal features
  - Phase-2: GBDT model with more features, including relevance and segmented popularity
  - Targeted explore/exploit, Diversity and variety, and Business rules

Federated Ranking

- Federated approach to solve different relevance problems
  - Flexibility to take advantage of multiple retrieval systems
    - Incrementally add new retrieval methods (highly clickable, social, personally relevant, contextual, CF….), which leads to faster improvements
  - Increase the throughput for experimentation
  - Faster iteration speed

- Major components
  - Feature server
  - NRT (near real time) data pipeline
  - Response prediction server
  - Federation layer

Feature Server

- Goal: Provide high-performance k/v storage for dynamic feature & content feature that served for variety ranking phase

- On-Prod Customers

Customers	Data & Usage
NRT	fetch doc features and store state record
Federation	fetch doc features for ranking, dedup & variety
Exploration	fetch NRT cfb events and store explored list

Near-Real-Time (NRT) Data Pipeline

- Receive the data stream (user_id, doc_id, event) and enrich a event with user profile and content features
  - Event: click, view/skip
- Compute features in high-dimension space
  - U_AG, U_CTY, U_YCT, U_WIKI
  - D_PUB, D_YCT, D_WIKI, D_UUID
  - C_UID_{PUB,YCT,WIKI}

Response Prediction Server

- Goal: scoring based on both user response feedback features and contextual features

- Methodology
  - Use historical performance and contextual features to predict future performance
  - Accumulate statistics at stable aggregate level instead of article level
  - It is essentially a smoothing method to tackle data sparseness

- Features
  - Response feedback features: multi-level CTR in different aggregate dimensions from NRT
  - Contextual features
  - Match-type specific features: match type score, confidence

Prediction Server – Signal Examples

- Event features

User CFB	Count
Age/Gender	30
City	16K
YCT	235
WIKI	171K
UID	10M

Doc CFB	Count
Publisher	210
YCT	255
WIKI	44K

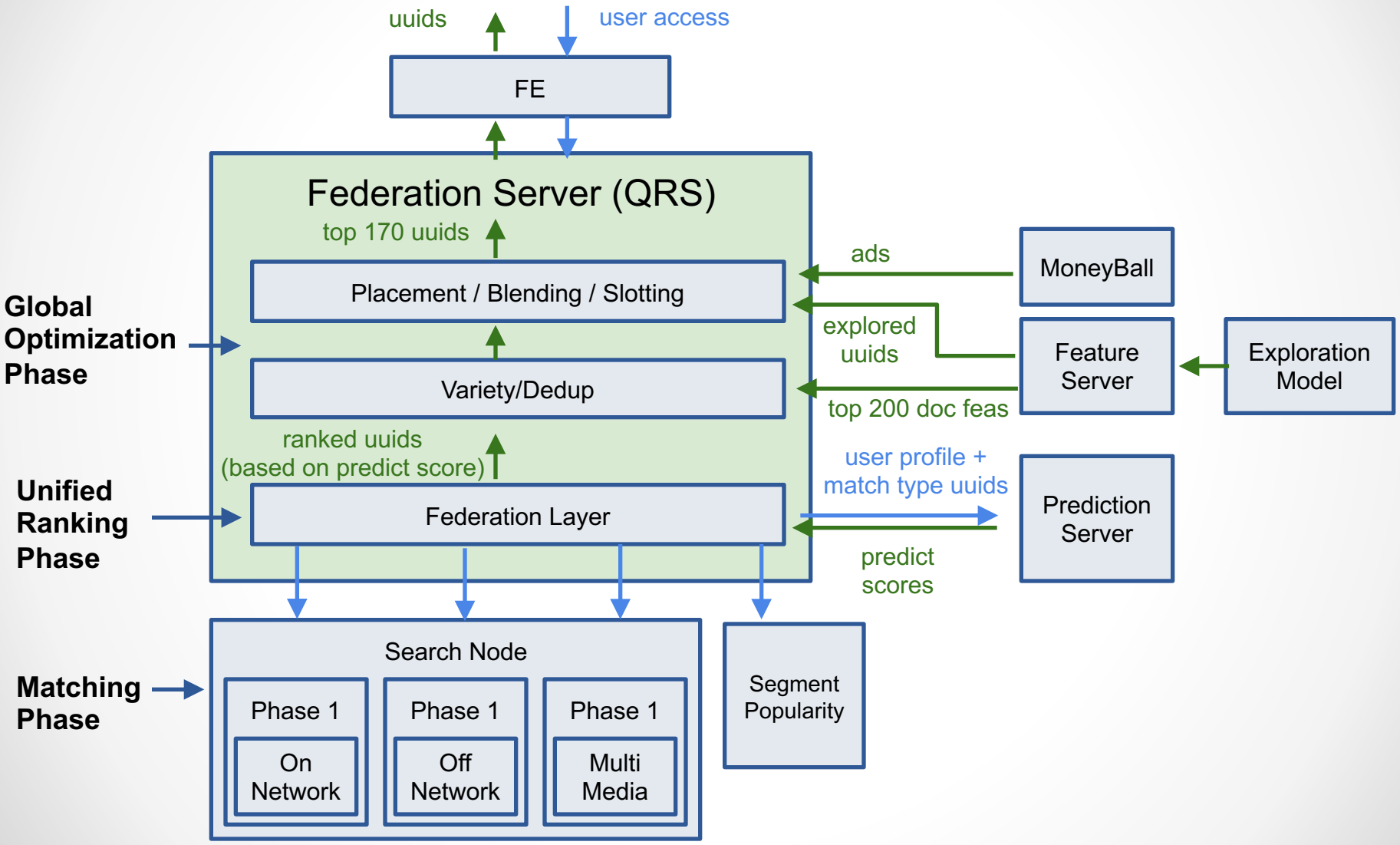
Cross CFB	Count
C_AG_{PUB,YCT,WIKI}	720K
C_CTY_{PUB,YCT,WIKI}	73M
C_YCT_{PUB,YCT,WIKI}	10M
C_WIKI_{PUB,YCT}	50M
M_WIKI	14K
C_UID_{PUB,YCT,WIKI}	3300M
TOTAL	3430M

- Non-event features
  - Popularity, quality, match type score, time-of-day, day-of-week

Federation Layer

- Goal: A glue layer to connect query and each matching types, call prediction server and features server to finish whole ranking process
- Physically embedded within Query Result server in search engine
- Multi-Match type supported
  - Search node, segmented popularity, social popularity etc.

Federated Ranking Framework



Thank You!

Appendix

Federated Ranking - Architect Diagram