Step 1: Outline Problem and Solution

Problem Statement:

Advertisers seek to personalize ads to individual users based on information known about them.

Challenges:

* Slow optimizations
* Many permutations
* Limited Reach
* Missed Opportunities
* Need to improve ROI

Solution:

Advertisers main goal of serving Ads is to obtain maximum reach and high optimization, this can be achieved by leveraging Dynamic Creative Optimization (DCO) system that helps with design, test and iteration across variations to find top-performing Ad combinations.

* DCO platform can be leveraged to deliver ad creatives @scale :

1) auto-generated dynamically
2) tested in real-time (A/B tests)
3) optimized for maximum clicks

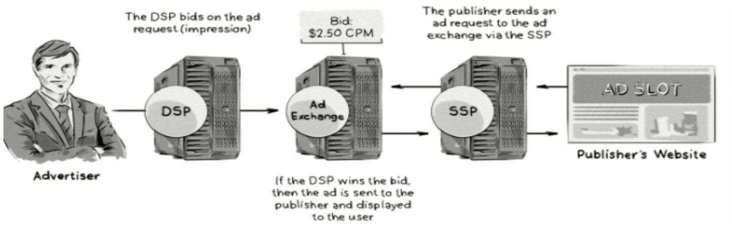
* Machine learning models can be as well leveraged and applied to predict which top-performing combinations (images, headlines, etc) will resonate most with target audiences.

Benefits:

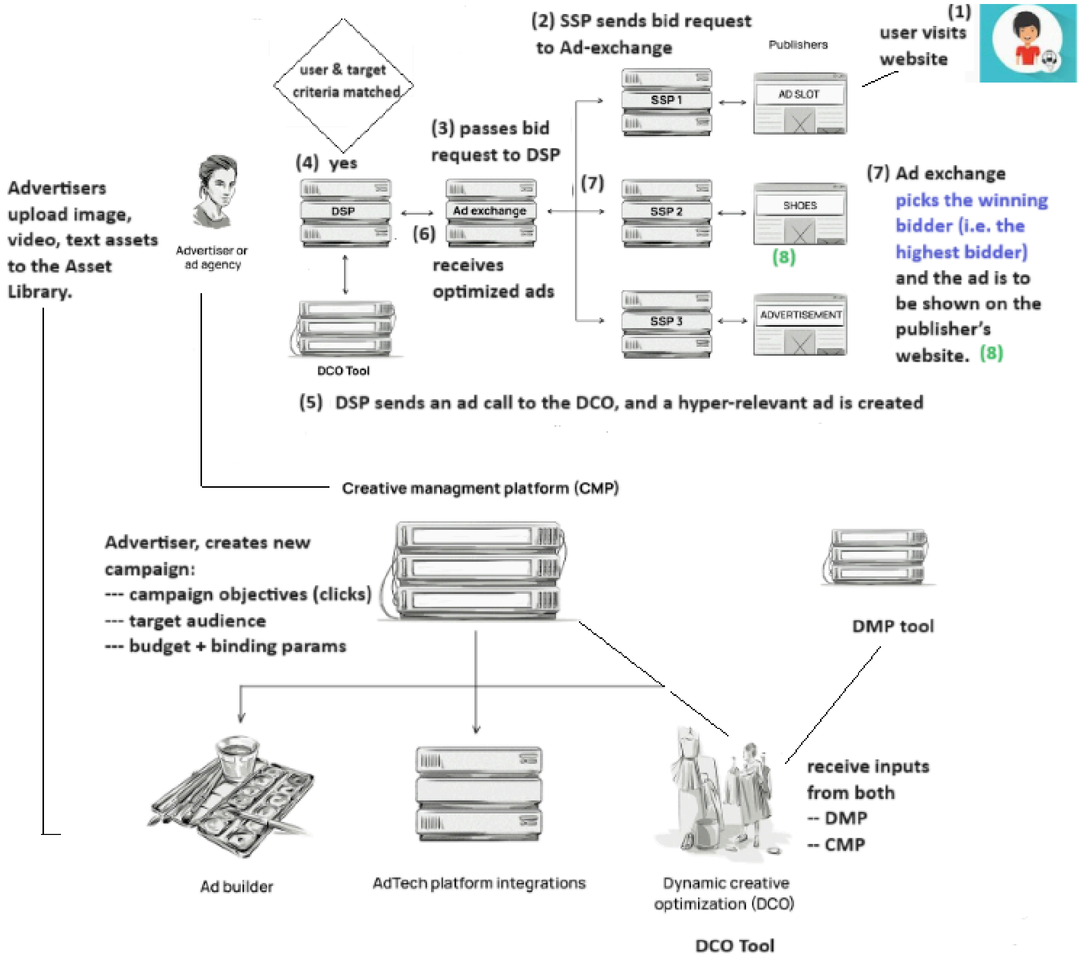
- Higher engagement
- Customized (*personalized) remarketing to each user
- Higher ROI
- Automation (automate your storyline across multiple products within same product line, save time, cost and increase efficiency of your display campaigns)

Simple Flow view

```
User Visits Website -->
SSP (Bid Request) --> Ad Exchange --> DSPs (Targeting & Bids)
                  --> Ad Exchange (Auction) --> Winning DSP (Ad Call)
                                                  --> DCO (Ad creation)
                  --> Ad Exchange --> Website (Ad Displayed)
```


The DSP bids on the ad request (impression)
Bid: $2.50 CPM
The publisher sends an ad request to the ad exchange via the SSP
DSP
Ad Exchange
SSP
AD SLOT
Publisher's Website
Advertiser
If the DSP wins the bid, then the ad is sent to the publisher and displayed to the user

Detailed Flow View : Dynamic Creative Optimization system



(1) user visits website

(2) SSP sends bid request to Ad-exchange

(3) passes bid request to DSP

(6) receives optimized ads

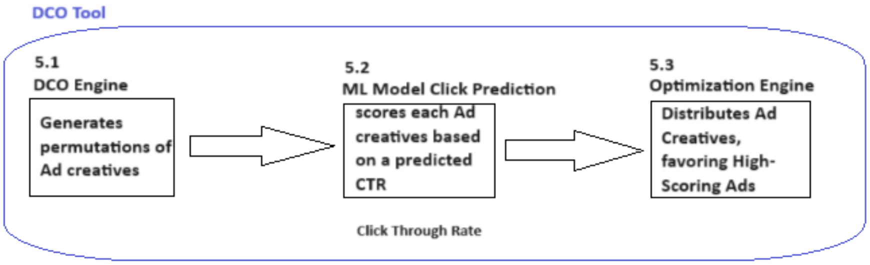
(7) Ad exchange picks the winning bidder (i.e. the highest bidder) and the ad is to be shown on the publisher's website. (8)

(5) DSP sends an ad call to the DCO, and a hyper-relevant ad is created

Advertisers upload image, video, text assets to the Asset Library.

Advertiser, creates new campaign:
--- campaign objectives (clicks)
--- target audience
--- budget + binding params

receive inputs from both
-- DMP
-- CMP



Generates permutations of Ad creatives

scores each Ad creatives based on a predicted CTR

Distributes Ad Creatives, favoring High-Scoring Ads

Step 2: Use Cases and Constraints

Use Cases

1> [UC01] User visits a website, and a new Ad it's published on the website for user consumption, due user information and Ad targeting criteria are a match. If no match, display fallback Ad.

2> [UC02] Save data from user (visitor) and latest ad bid request, to enrich and meet other business goals

3> [UC03] Fine tune - Click Prediction Models + optimize high scored ads, from latest saved data

4> [UC04] Ability to analyze all data and make it available to external tools and dashboards as required

[UC01]

- **User** a user visits a website
- A supply-side platform **(SSP)** sends a bid request to an Ad-exchange
- **Ad-exchange** passes bid request to all the DSPs it integrates with
- **Each DSP** needs to respond to a bid request
  - Evaluate if known user info (criteria) **matched** with the targeting criteria > one campaign
    - Return success bid response :: Highest bid (**winner bid**)
    - Return success bid response :: Medium - low bid (loser bid)
  - Evaluate if known user info (criteria) did **not matched** any targeting criteria
    - Return failed bid response :: NoMatchFound

- **Ad-exchange** process Return bid (auction)
  - Bid-response => Failed-NoMatchFound
    - Ad-exchange returns to SSP - no ad created

  - Bid-response => Success-LoserBid
    - Ad-exchange returns to SSP - no ad created

  - Bid-response => Success-WinnerBid
    - **DSP** makes a call to **DCO tool**
      - **DCO tool** returns to DSP - a hyper-relevant ad created

- **DSP** returns to Ad-exchange - new ad created (optimized | highest scored)
- **Ad-exchange** returns to SSP - new ad created

- **SSP** process response from Ad-exchange
  - Response => No ad created
    - Nothing is shown in publisher website
  - Response => New ad was created
    - New Ad is pushed and shown in the publisher website

- [UC03] **DCO tool** receives request to generate Ad from DSP
  - DCO engine generates permutations of Ad creatives
    - Ads creatives generated are sent for scoring to ML model

  - ML model (Click Prediction)
    - Each Ad creative gets scored based on predictive CTR
      - Ad + score is sent for prioritization to optimization engine

  - Optimization engine
    - Distributes Ad creatives and spend, and determines high-score Ad
      - Highest scored Ad is returned

[UC02]

- [1] **User** information and latest Ads requests are saved in batch as csv file in amazon Kinesis and stored in S3 buckets.

[UC04] Ability to analyze all data and make it available to external tools and dashboards as required

DCO additional cases

Location-based customization: You can modify creatives to match regional preferences or details about the company’s local offerings. The same applies to demographic and behavioral variables.

Retargeting campaigns and promotions: You can encourage consumers to convert by dynamically optimizing ads with down-sales, cross-sales, and special promotions. You can also show your ads to consumers who expressed intent in buying your products (e.g. abandoned a shopping cart and visited a particular product page)

Contextual targeting: You can modify creatives to be more appealing to visitors of specific pages, allowing the ads to deliver a seamless user experience.

Constraints and assumptions (*)

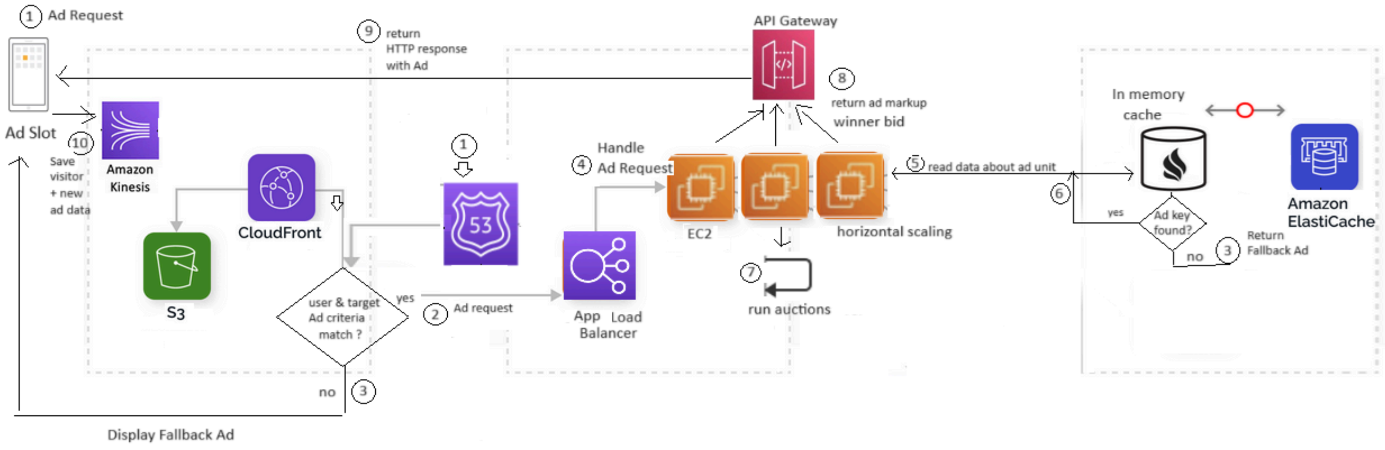
- The system must handle up to 1k ad requests/sec
  * system needs to be efficient; handle multi-processing | highly scalable
- Latency per request should be < 50ms
  * system needs to be as fast as possible + handle caching + or CDN to handle content replication - so Ads are displayed faster per geographical area
  * search operations in DB’s in millions of records - DB should handle powerful indexes
- The system should handle up to 1 million ad creative options at any given time
  * Peak traffic load - scenario
- There are ~10 million users on the platform

Additional Questions for the Interviewer?

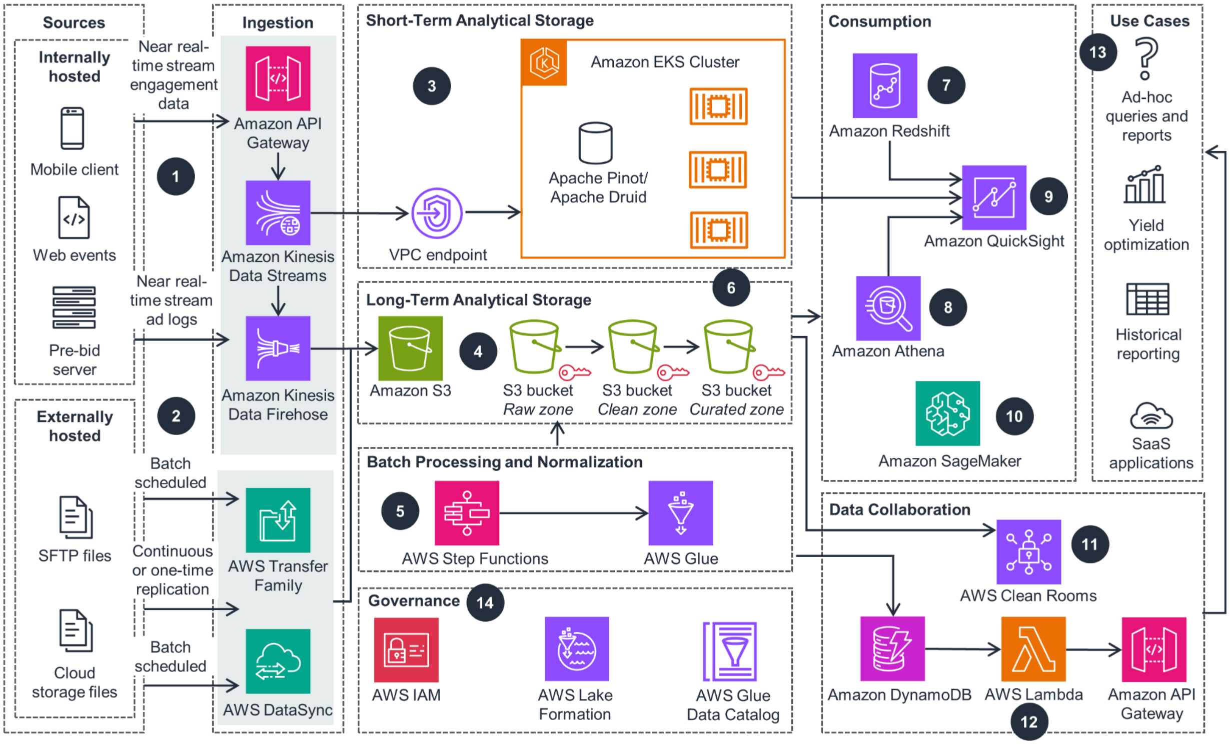
- Should we account for traffic that is not evenly distributed ?
- Do we need to consider adaptability to rapid changes in demand?
- What would be the potential growth in terms of users base or volume of data to process?
- What are components to revisit in the future due big data/extreme growth?

Step 3: System Architecture

Simple Architecture



More Robust Architecture - AWS



Key Components - Simple Architecture

This section relates primarily to the visitor and the decision-making process that goes into showing them an ad when they visit a web page.

Supply Side Platform, Ad exchange

When a web page starts loading, information about the visitor (geolocation, device model, etc.) is passed to the ad exchange either by collecting it directly from the publisher's website or via a supply-side platform (SSP).

Targeted ads are computed on-the-fly based on cookie or query string data, and advertisers generally need low latency in serving ads.

- **Amazon CloudFront and Amazon Route 53**

  When a user visits a web page (e.g. http://example.com), an Ad request will be triggered and static content (Ads) will be fetched.

  Benefits:

  Route 53 helps with routing end users by running multiple endpoints (globally), driving traffic flow to the best endpoint on latency, geography and endpoint health.

  With Amazon CloudFront, end users' connections are terminated at CloudFront locations closer to them, which helps in reducing the overall round trip time required to establish a connection.

  Amazon CloudFront can help meet the performance and personalization needs for such Ads applications either by accelerating ad targeting calls from client to ad server, caching and delivering the ad creatives, or optimizing the reporting beacon calls.

  Trade offs: High cost at scale CloudFront pricing is pay-per-use

- **Application Load Balancer, EC2**
  If known user information and the Ad targeting are a match, then configured DNS record points the request to the load balancer which in turn sends a request to EC2 instances

  Benefits: increases the availability and fault tolerance of your applications.
  Trade Offs: number of instances that can be running at a time

- **ElasticCache, In-Memory cache**
  EC2 makes a request for fetch data of Ad from In-memory cache db and if the key provided is found, then ad data is returned to EC2.

  If the Ad key is not found, there is an option to fallback directly at ElasticCache db, but if still, after this the key remains not found. When no AdUnitID is found, some fallback ad is sent back and displayed to the visitor.

  Benefits: In-memory caching improves application performance by storing critical pieces of data in memory for low-latency access

  Trade offs: ElastiCache doesn't support autofailover from one AWS Region to another

DSP (Bids)

- When AdUnitID is found, an EC2 instance executes an auction between defined vendor(s) — typically DSPs and ad networks — in a limited amount of time.This can be one vendor or multiple vendors.

API Gateway

- The winning vendor then sends the ad markup to the browser and displays the ad to the visitor.

  Benefits:
  It allows for efficient distribution of traffic and ensures proper load balancing among target endpoints. Simplified routing and request handling

  Trade offs:
  Can become a bottleneck if not properly managed.

Kinesis

- While the ad is being displayed, all information gathered about the visitor and the ad itself are passed to Kinesis for further processing.

Benefits: Allows to process and analyze data nearly immediately.
Cost-effectively processes and analyzes streaming data at any scale as a fully managed service

Trade offs: Number of shards per stream limited, rate limited and others.

Key Components - Detailed Flow View

[Static component] - constant or unchanged throughout the DCO process
[Dynamic component] - adapt and personalize ad content based on real time data

[Static component]

Data Management Platform (DMP)

Data Feeds:
Real-time data sources informing the DCO engine
(e.g., location, weather, preferences)

goals:
>> provide real time data about users behavior, preferences and demographics
>> collect first party, second party, third party data
>> segment audiences based on demographics, interests, behavior, location

[Static component]

Creative Management Platform (CMP)

Asset Library
* Repository for storing a diverse range of creative assets
Images (various formats, sizes)
Videos (as needed)
Headlines (multiple variations)
Body Copy (multiple variations)
Calls to Action (CTAs)

* Metadata tagging for categorization and filtering assets
(e.g., product type, themes, target demographics)

goals:
>> store creative assets
>> store creative templates or rules defining how assets can be combined
>> store and manages ad creatives for different audience segments

[Dynamic component]

DCO Engine

Decision-making component of the system:

* Dynamically assembles ad creatives by combining static elements (CMP) with personalized data (DMP) and generates permutations

goals:

>> compose combinations of assets from the library

>> apply predefined templates and design rules

>> enforce ad network technical specifications

[Dynamic component]

Machine Learning Model (Click Prediction)

Trained on historical campaign data

Visual characteristics of assets

Textual content, sentiment, keywords

Target audience attributes

Past ad performance metrics

Predicts the Click-Through-Rate (CTR) of each generated ad creative

[Dynamic component]

Optimization Engine

2nd core decision-making component of the system:

* Drives prioritization of well-performing creatives

* Assess and manage retirement of underperforming creatives

* Applies appropriate optimization algorithms

* Helps with allocation of ad spend across variations

[Dynamic component]

Demand-Side Platform (DSP):

>> Receives optimized ads from the DCO engine for bidding

>> Demand-side platform where the ad buying process takes place

>> Seamless API integration with major ad networks (Google Ads, Meta, etc)

>> Stores and manages campaigns, ad sets, individual ad creatives, etc.