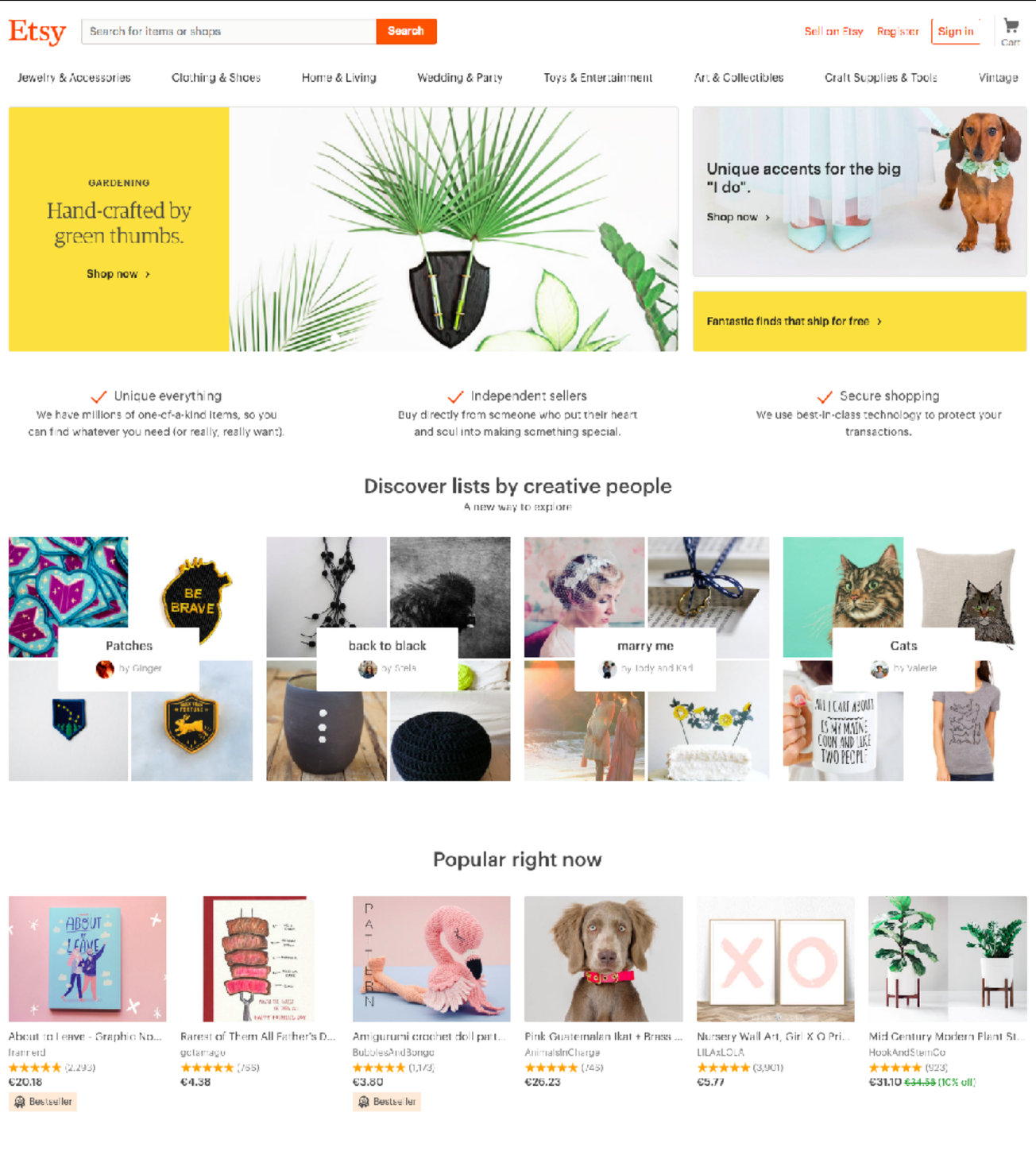
Aakash Sabharwal
@aakashsabharwal

Stefanie Schirmer
@linse

Your Search Service as a Composable Function

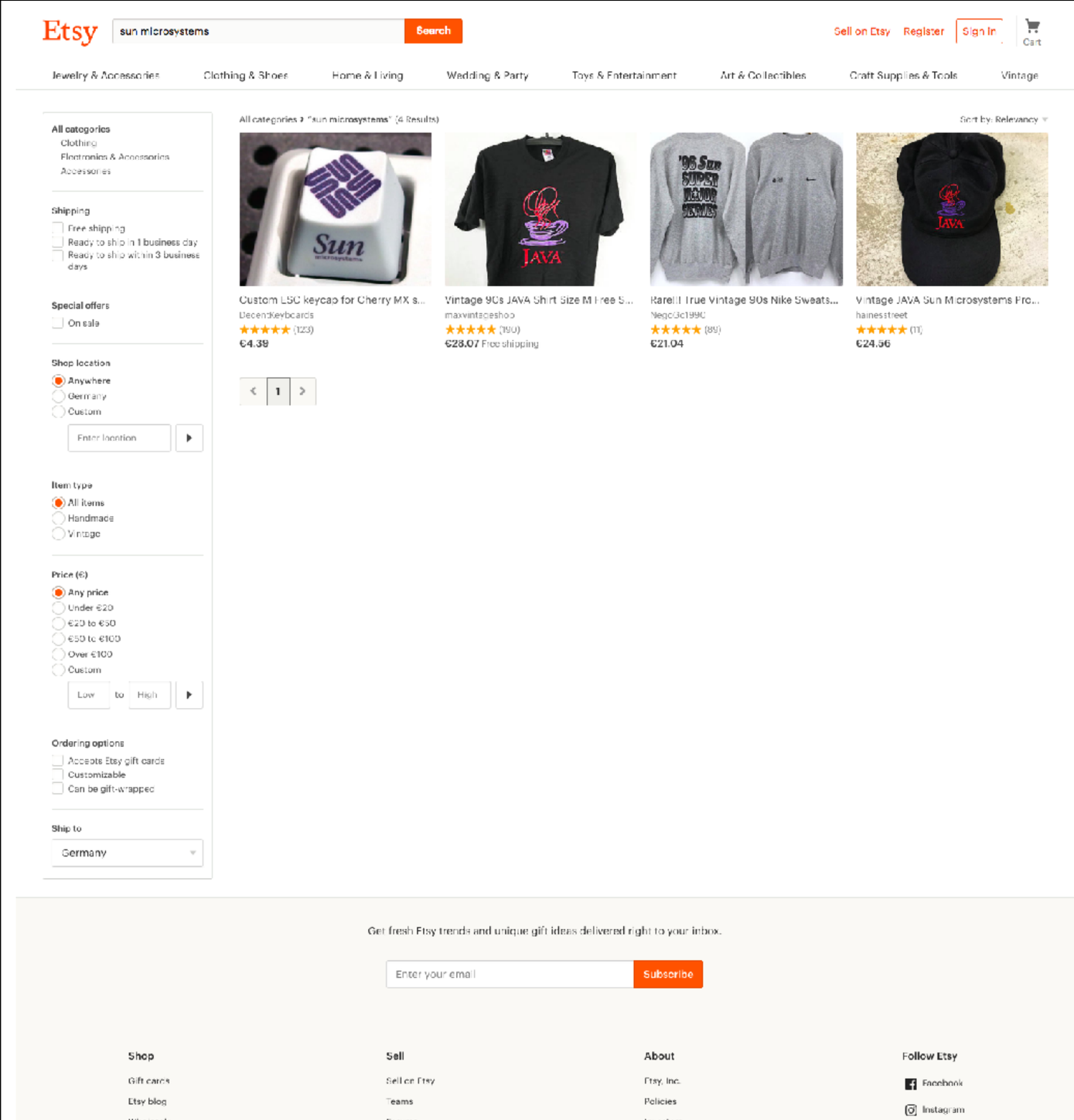
Etsy

- online marketplace
- 1.5 million active sellers
- handmade + vintage items



Etsy

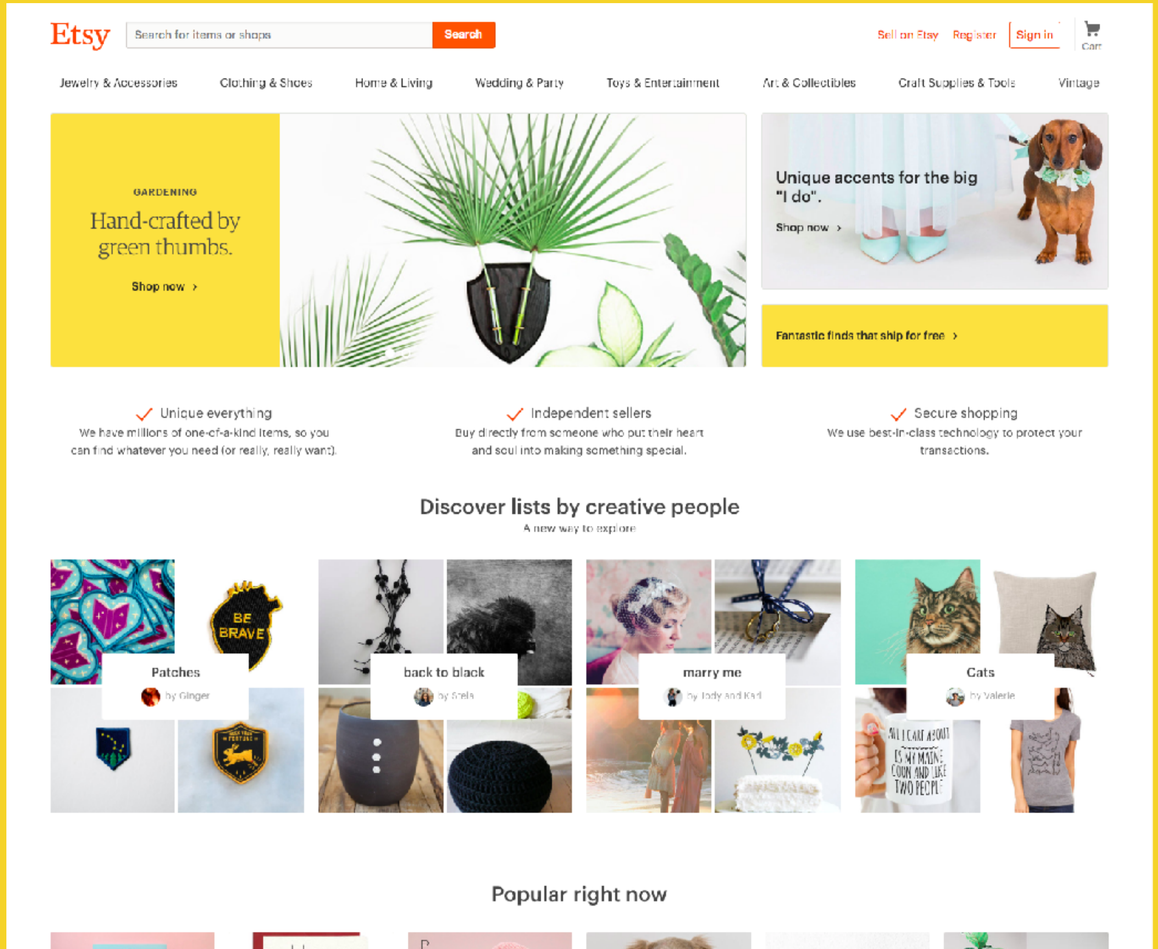
- online marketplace
- 1.5 million active sellers
- handmade + vintage items
- unique search problem!

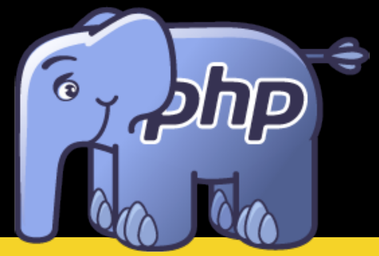


Infrastructure Challenge

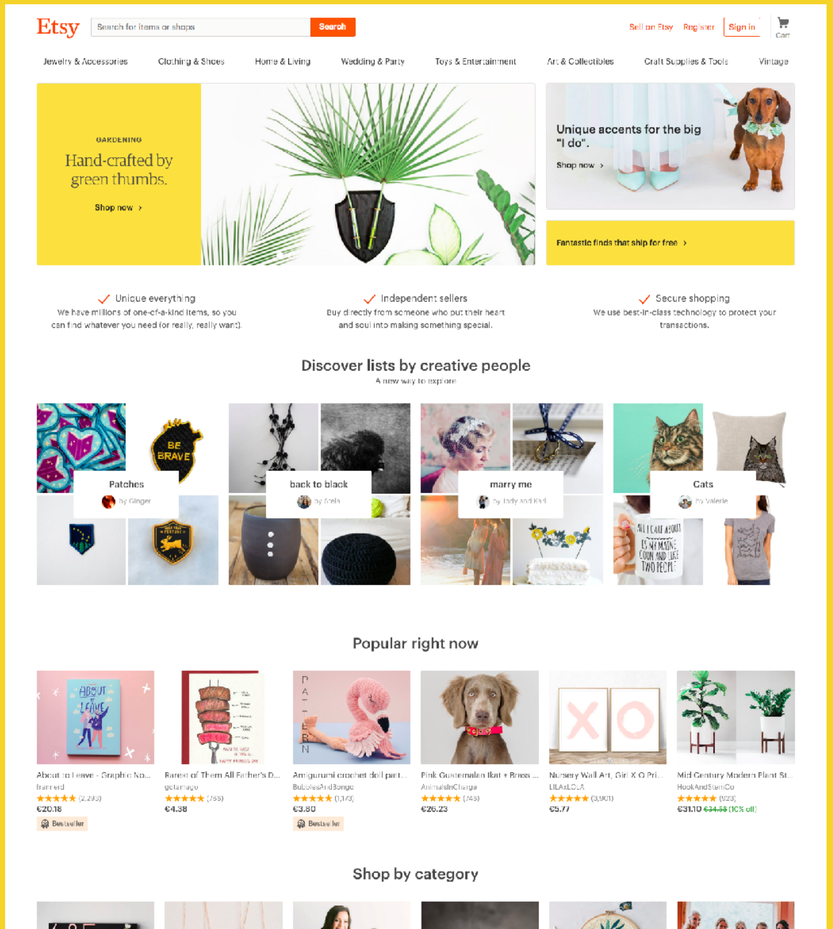
php

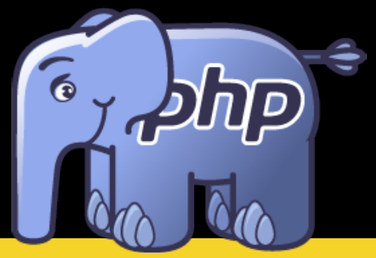
Etsyweb


Etsy
Search for items or shops
Search
Sell on Etsy
Register
Sign In
Cart
Jewelry & Accessories
Clothing & Shoes
Home & Living
Wedding & Party
Toys & Entertainment
Art & Collectibles
Craft Supplies & Tools
Vintage
GARDENING
Hand-crafted by green thumbs.
Shop now ›
Unique accents for the big "I do".
Shop now ›
Fantastic finds that ship for free ›
Unique everything
We have millions of one-of-a-kind items, so you can find whatever you need (or really, really want).
Independent sellers
Buy directly from someone who put their heart and soul into making something special.
Secure shopping
We use best-in-class technology to protect your transactions.
Discover lists by creative people
A new way to explore
Patches
by Ginger
back to black
marry me
Cats
by Valerie
Popular right now

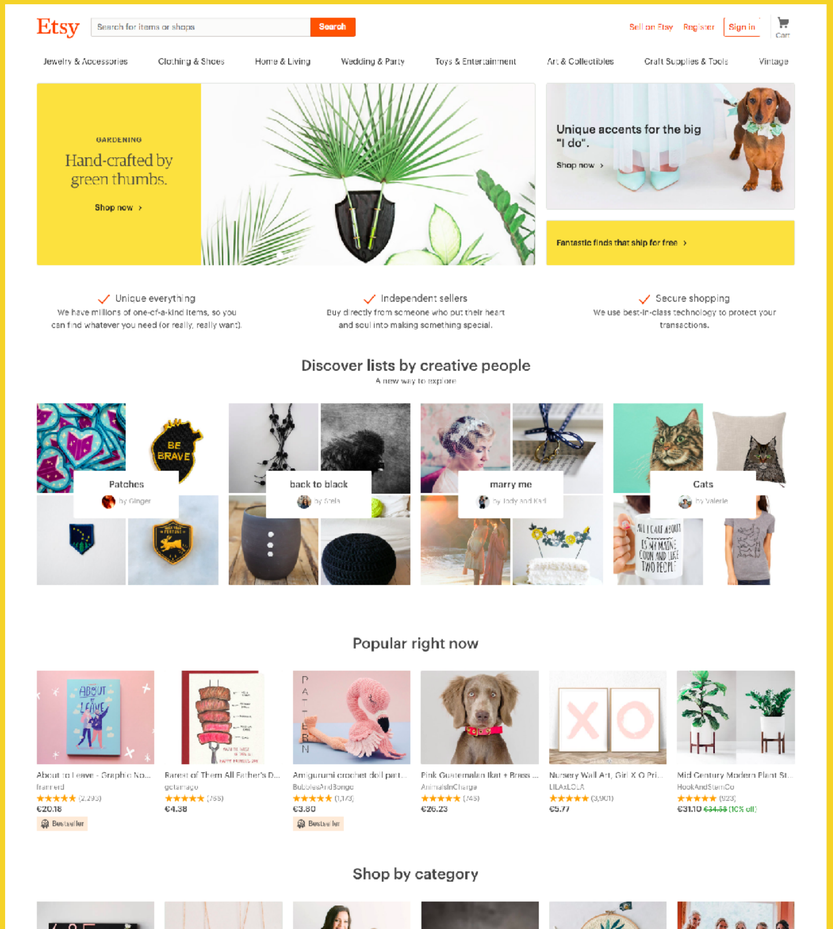

php

Etsyweb

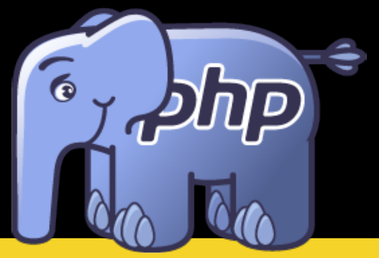

Etsy
Search for items or shops
Search
Sell on Etsy
Register
Sign in
Cart
Jewelry & Accessories
Clothing & Shoes
Home & Living
Wedding & Party
Toys & Entertainment
Art & Collectibles
Craft Supplies & Tools
Vintage
GARDENING
Hand-crafted by green thumbs.
Shop now
Unique accents for the big "I do".
Shop now
Fantastic finds that ship for free
Unique everything
We have millions of one-of-a-kind items, so you can find whatever you need (or really, really want).
Independent sellers
Buy directly from someone who put their heart and soul into making something special.
Secure shopping
We use best-in-class technology to protect your transactions.
Discover lists by creative people
A new way to explore
Patches
back to black
marry me
Cats
Popular right now
Shop by category



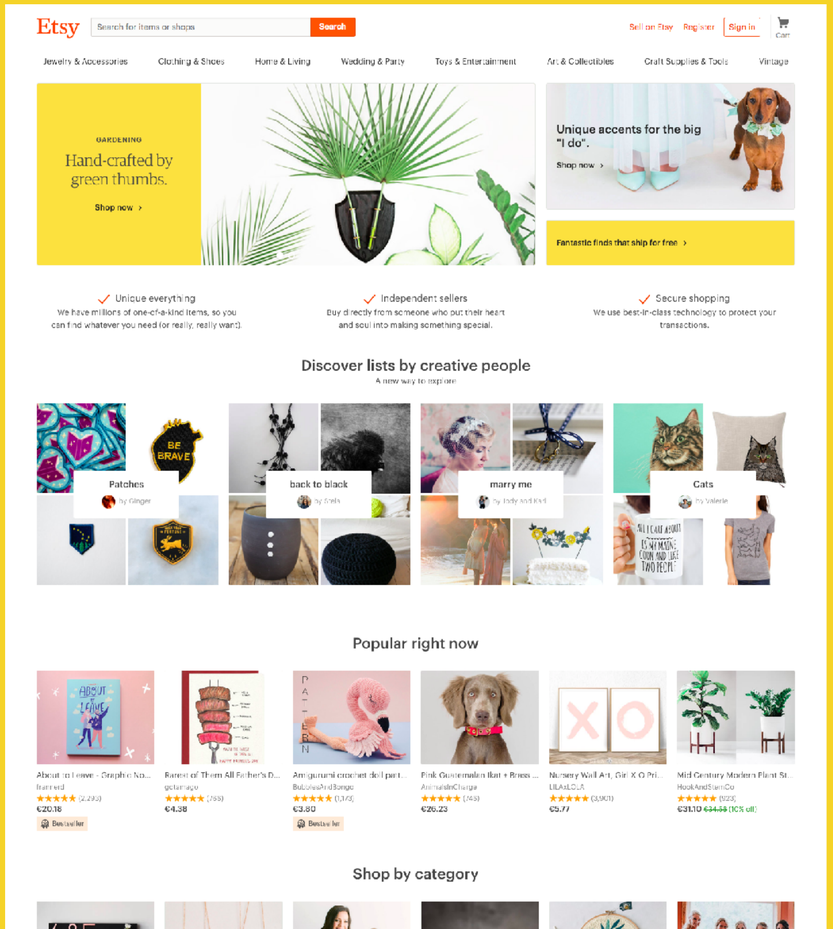
Etsyweb



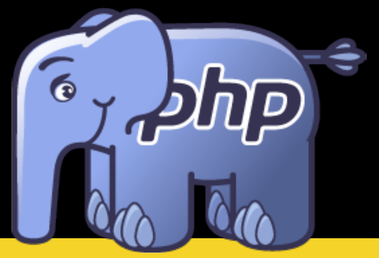
Search Backend



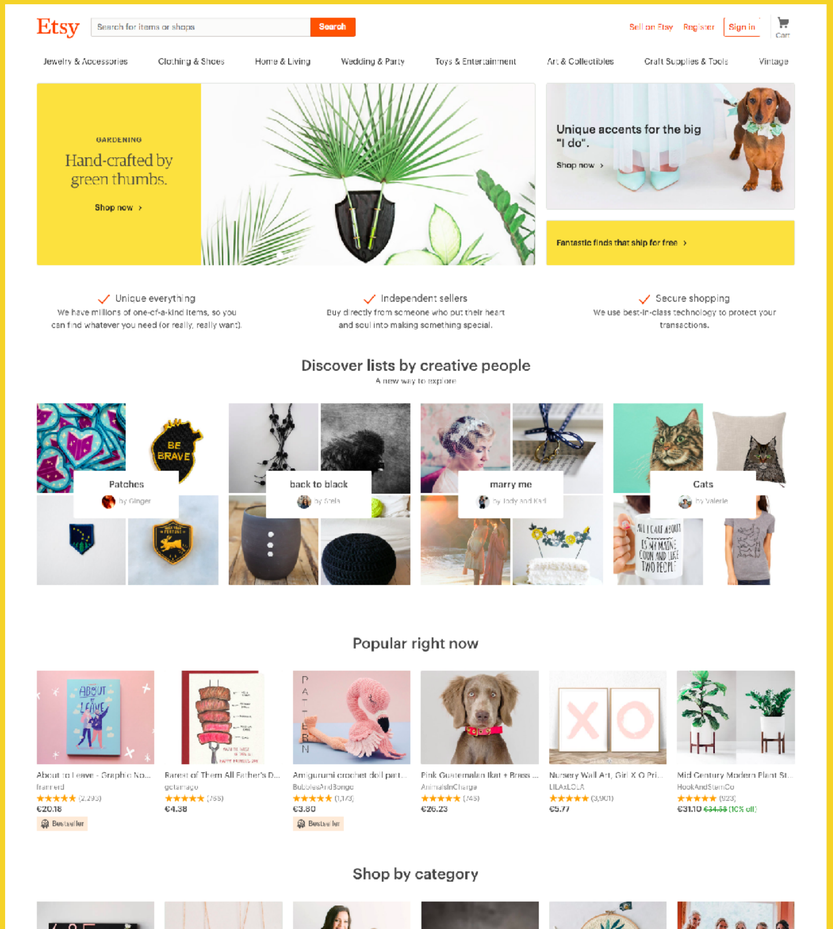
Etsyweb



Search Backend

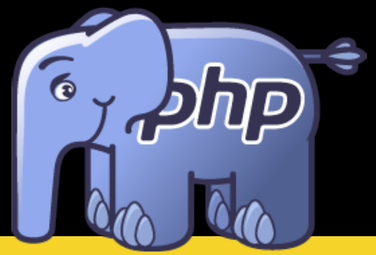


Etsyweb

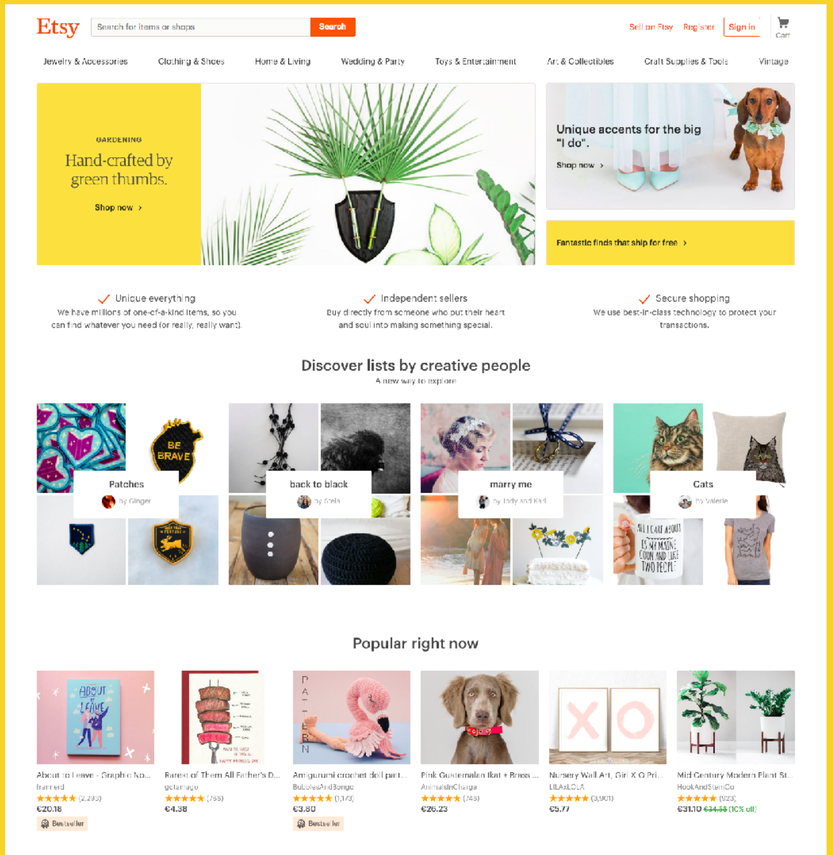


Search Backends

Machine Learning Models

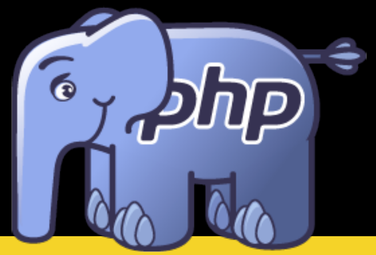


Etsyweb

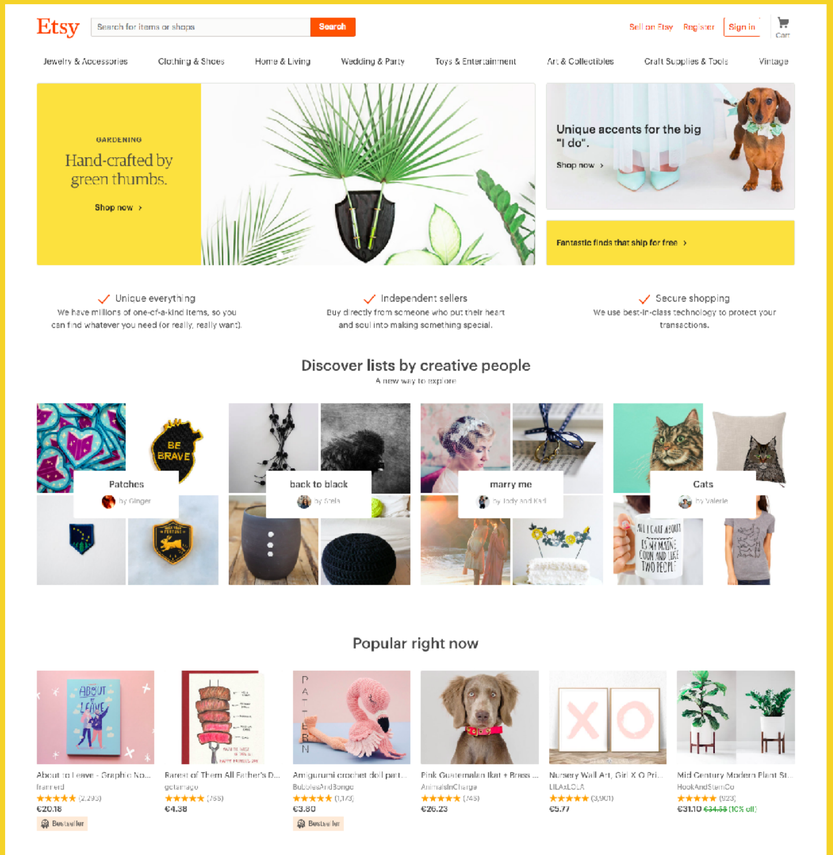


Search Backends

Machine Learning Models



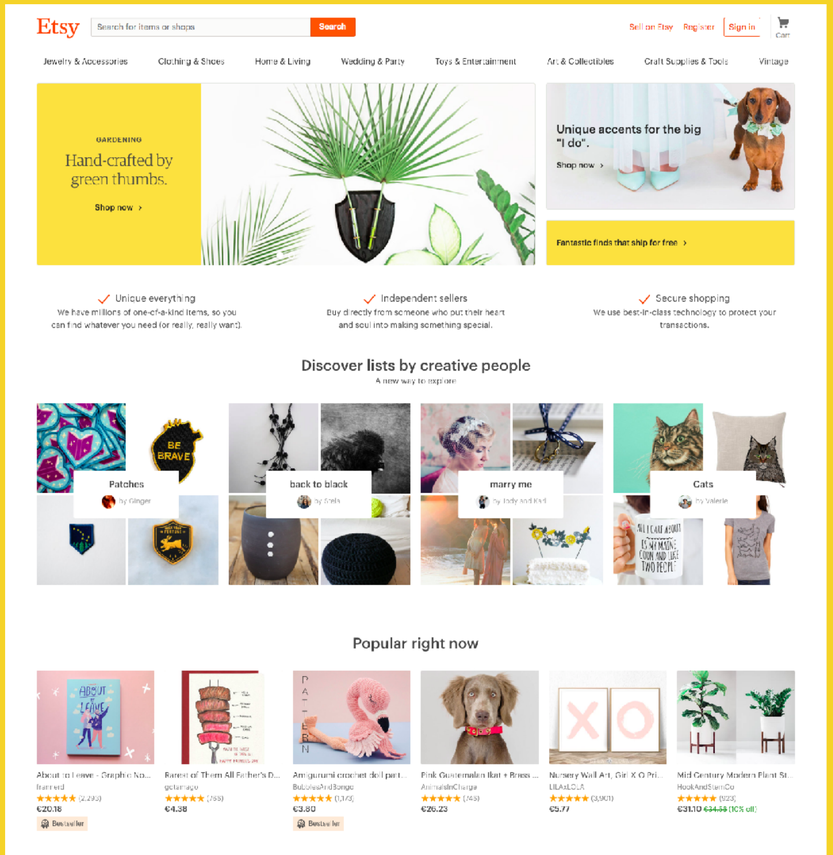
Etsyweb



Search Backends

Machine Learning Models

Etsyweb



Search Backends

Acquiring a New Approach

Etsy buys Blackbird Technologies to bring AI to its search

Ingrid Lunden @ingridlunden / Sep 20, 2016

Comment

It's not just gigantic search engines that are looking to acquire tech and talent around emerging areas like speech recognition. Even design and craft marketplaces can use a little machine learning and artificial intelligence to make their wheels turn a little better.

Today, the popular handmade-goods site Etsy announced it has acquired a startup called Blackbird Technologies, which developed algorithms for natural language processing, image recognition and analytics — similar to those used by Amazon and Google for product and other searches — and then "democratized" them to be used by any company of any size. At Etsy, the tech will be used to improve its own search features.

Financial terms of the deal have not been disclosed, but we're asking and will update if we learn more. The company's employees — it appears there were around 10 — will all be joining Etsy. "The team possesses a deep expertise in Artificial Intelligence, search, and distributed systems and has direct experience working in these areas for some of the largest technology companies in the world," **Etsy** ◍ said in a statement.

Little is known about Blackbird — no listings on Crunchbase or AngelList, a fairly unspecific website for Blackbird itself — and even Etsy's announcement of the deal is light on company details. According to LinkedIn, though, the two co-founders are Nikhil Raghavan (CEO) and Manju Rajashekhar (CTO), who collectively had worked on similar technologies across places like Google, Twitter, Microsoft, Oracle and Yahoo.

Because Etsy is going to be using Blackbird's tech for its own platform, it seems that Blackbird's current business plans, which deployed its tech by way of an API, will shut down. Customers had included Nasty Gal,

Functional Programming in

Scala

Paul Chiusano
Rúnar Bjarnason

MANNING

Your Server as a Function

Marius Eriksen
Twitter Inc.
marius@twitter.com

Abstract

Building server software in a large-scale setting, where systems exhibit a high degree of concurrency and environmental variability, is a challenging task to even the most experienced programmer. Efficiency, safety, and robustness are paramount—goals which have traditionally conflicted with modularity, reusability, and flexibility.

We describe three abstractions which combine to present a powerful programming model for building safe, modular, and efficient server software: Composable *futures* are used to relate concurrent, asynchronous actions; *services* and *filters* are specialized functions used for the modular composition of our complex server software.

Finally, we discuss our experiences using these abstractions and techniques throughout Twitter's serving infrastructure.

Categories and Subject Descriptors D.1.1 [*Programming techniques*]: Applicative (Functional) Programming; D.1.3 [*Program-*

Services Systems boundaries are represented by asynchronous functions called *services*. They provide a symmetric and uniform API: the same abstraction represents both clients and servers.

Filters Application-agnostic concerns (e.g. timeouts, retries, authentication) are encapsulated by *filters* which compose to build services from multiple independent modules.

Server operations (e.g. acting on an incoming RPC or a timeout) are defined in a declarative fashion, relating the results of the (possibly many) subsequent sub-operations through the use of future combinators. Operations are phrased as *value transformations*, encouraging the use of immutable data structures and, we believe, enhancing correctness through simplicity of reasoning.

Operations describe *what* is computed; execution is handled separately. This frees the programmer from attending to the minu-

Your Server as a Function

Set of Composable

Req => Future[Rep]

Marius Eriksen
Twitter Inc.
marius@twitter.com

Abstract

Building server software in a large-scale setting, where systems exhibit a high degree of concurrency and environmental variability, is a challenging task to even the most experienced programmer. Efficiency, safety, and robustness are paramount—goals which have traditionally conflicted with modularity, reusability, and flexibility.

We describe three abstractions which combine to present a powerful programming model for building safe, modular, and efficient server software: Composable *futures* are used to relate concurrent, asynchronous actions; *services* and *filters* are specialized functions used for the modular composition of our complex server software.

Finally, we discuss our experiences using these abstractions and techniques throughout Twitter's serving infrastructure.

Categories and Subject Descriptors D.1.1 [*Programming techniques*]: Applicative (Functional) Programming; D.1.3 [*Program-*

Services Systems boundaries are represented by asynchronous functions called *services*. They provide a symmetric and uniform API: the same abstraction represents both clients and servers.

Filters Application-agnostic concerns (e.g. timeouts, retries, authentication) are encapsulated by *filters* which compose to build services from multiple independent modules.

Server operations (e.g. acting on an incoming RPC or a timeout) are defined in a declarative fashion, relating the results of the (possibly many) subsequent sub-operations through the use of future combinators. Operations are phrased as *value transformations*, encouraging the use of immutable data structures and, we believe, enhancing correctness through simplicity of reasoning.

Operations describe *what* is computed; execution is handled separately. This frees the programmer from attending to the minu-

Machine Learning Models

Search Backends

Etsyweb

Goal: rewrite search query + response

Solr

elasticsearch

Machine Learning Models

Search Backends

Etsyweb

Format:

proxy, “no state”

async + protocol agnostic

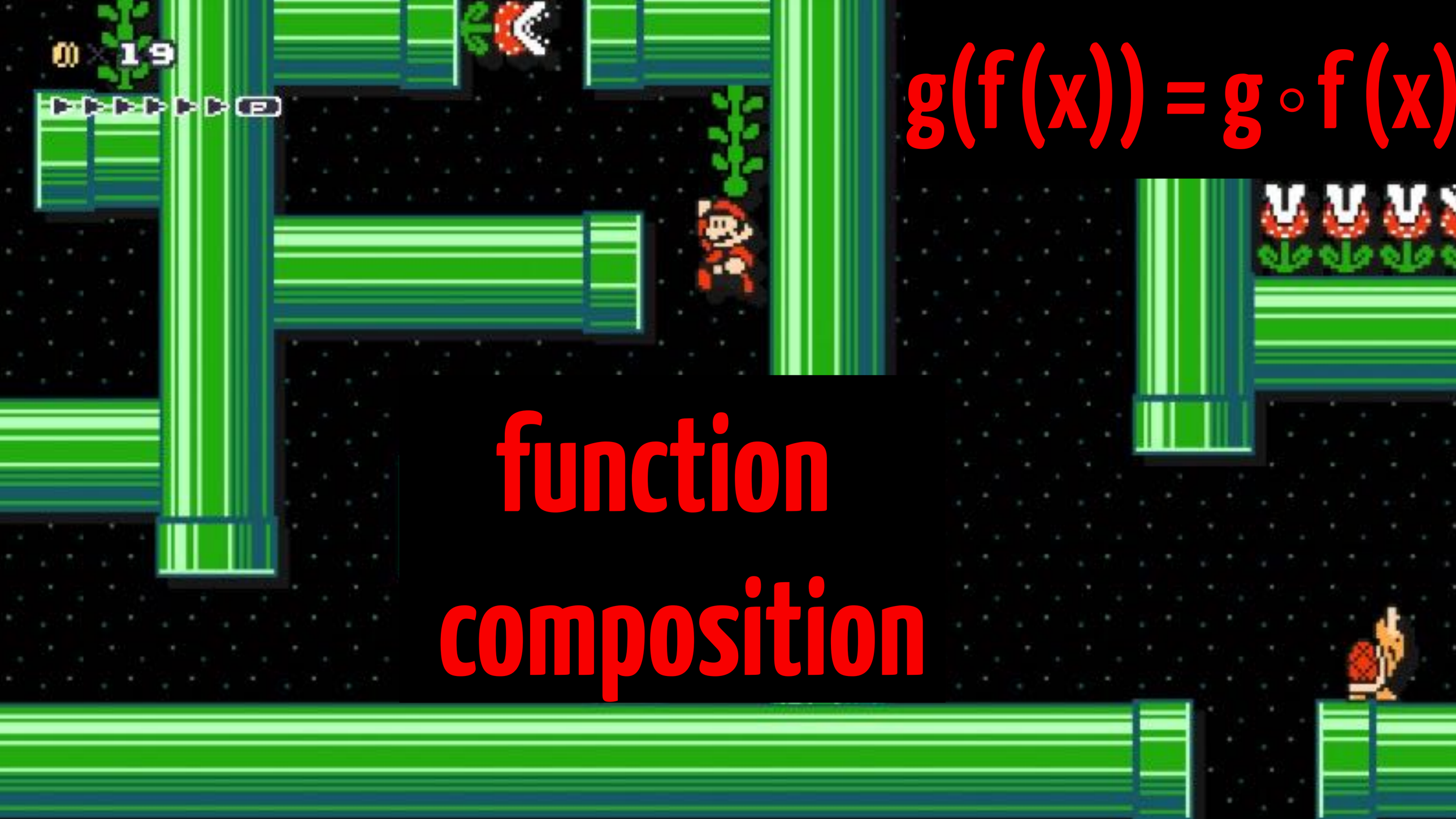
Solr

elasticsearch

BACK
TO THE FUTURE
TM

Future + slow call

Service + slow service


$g(f(x)) = g \circ f(x)$
function
composition

Service with filter

```
import com.twitter.finagle.Service
import com.twitter.finagle.http

val service: Service[http.Request, http.Response] = ...
val timeoutFilter = new TimeoutFilter[http.Request, http.Response](...)

val serviceWithTimeout: Service[http.Request, http.Response] =
  timeoutFilter.andThen(service)
~
~
~
~
~
~
~
~
~
~
-- VISUAL --
```

A Finagle Service

```
trait Service[Req, Rep]
extends (Req => Future[Rep])
```

Example: Service[ArizonaReq, ArizonaRsp]

```
import com.twitter.util.Future
import org.jboss.netty.handler.codec.http._
import com.twitter.logging.Logger
import com.twitter.finagle.http.{Method, Request, Response, Version}
import com.twitter.finagle.{Service, SimpleFilter}

import scalaz.{\/, \/-}
import java.util.zip.CRC32

object ArizonaService {
  // Usually we read this map from production.php. This is for testing convenience only.
  val shard2hostForTestingOnly = Map(1 -> "az-shard-1-vip.ny2.etsy.com:80", 2 -> "az-shard-2-vip.ny2.etsy.com:80")
}

case class ArizonaService(name: String, shard2host: Map[Int, String], sro: Option[StatsReceiver], timeout: Int = 100)
  extends Service[ArizonaRequest, ArizonaResponse] {

  private val log         = Logger(getClass)
  private val sr          = sro.getOrElse(DefaultStatsReceiver)
  private val retryFilter = RetryTimeoutsFilter[Request, Response](sr, log, 1)

  if (shard2host.isEmpty) {
    throw new Exception("shard2host map was empty.");
  }

  private[this] val shard2client = shard2host.map { case (s, h) => (s, shardedClient(s, h)) }.toMap

  private[this] val threadLocalShardHasher = new ThreadLocal[CRC32] {
    protected override def initialValue(): CRC32 =
      new CRC32
  }

  def shardedClient(shard: Int, host: String): Service[Request, Response] = {
```

apply: ArizonaReq => Future[ArizonaRsp]

```
def apply(req: ArizonaRequest): Future[ArizonaResponse] =
  req match {
    case MultiGet(_, keys) if keys.size > MultiGet.maximumRequestSize =>
      Future.exception(
        new RuntimeException(
          s"MultiGet support up to MultiGet.maximumRequestSize per request; got ${keys.size}"
        )
      )
    case _ => {
      val datasetName = req.dataset_name
      val shardHasher = threadLocalShardHasher.get()
      shardHasher.reset()
      shardHasher.update(datasetName.getBytes(StandardCharsets.UTF_8))
      val shardNum = (shardHasher.getValue() % shard2client.keys.size).toInt + 1
      val client   = shard2client.getOrElse(shardNum, throw new Exception(s"Client for shard $shardNum missing."))

      val r = Request(Version.Http11, Method.Get, makeUrl(req))
      r.headerMap.set(HttpHeaders.Names.CONNECTION, HttpHeaders.Values.KEEP_ALIVE)
      r.headerMap.set(
        HttpHeaders.Names.HOST,
        shard2host.getOrElse(shardNum, throw new Exception(s"Host for shard $shardNum missing."))
      );

      client(r).map { resp =>
        makeResponse(req, resp)
      }
```

fail if multiget #keys exceeded

find client for shard

set request headers

```
Service: Req => Future[Rep]
```

```
def Future[A].flatMap[B](f: A =>
    Future[B]): Future[B]
```

Future: can flatMap over it!

```scala
val f = for {
  user   <- authenticate(request)
  banned <- isBanned(user)
} yield (user, banned)
```

reader monad

Readers

**/ListingSearchThriftController.scala:

```
def handle(Search) { args: Search.Args =>
  val req = constructReq(args)
  val rsp = for {
    config   <- cdb.read[Config](req)
    index    <- req.read("index") \/> "url path is missing index component"
    ctx      <- config.read[SearchCSRCtx](index -> req)
  } yield {
    searchWithBackoff(args)(ctx.listingsCtx)
  }
  rsp.fold(e => Future.exception(new RuntimeException(e)), resFuture => resFuture)
}
```

A Merlin Mixer Service

```
Req => ReaderToFuture[Ctx, Rep]
```

Type: Reader to Future

ReaderMonad + FutureMonad

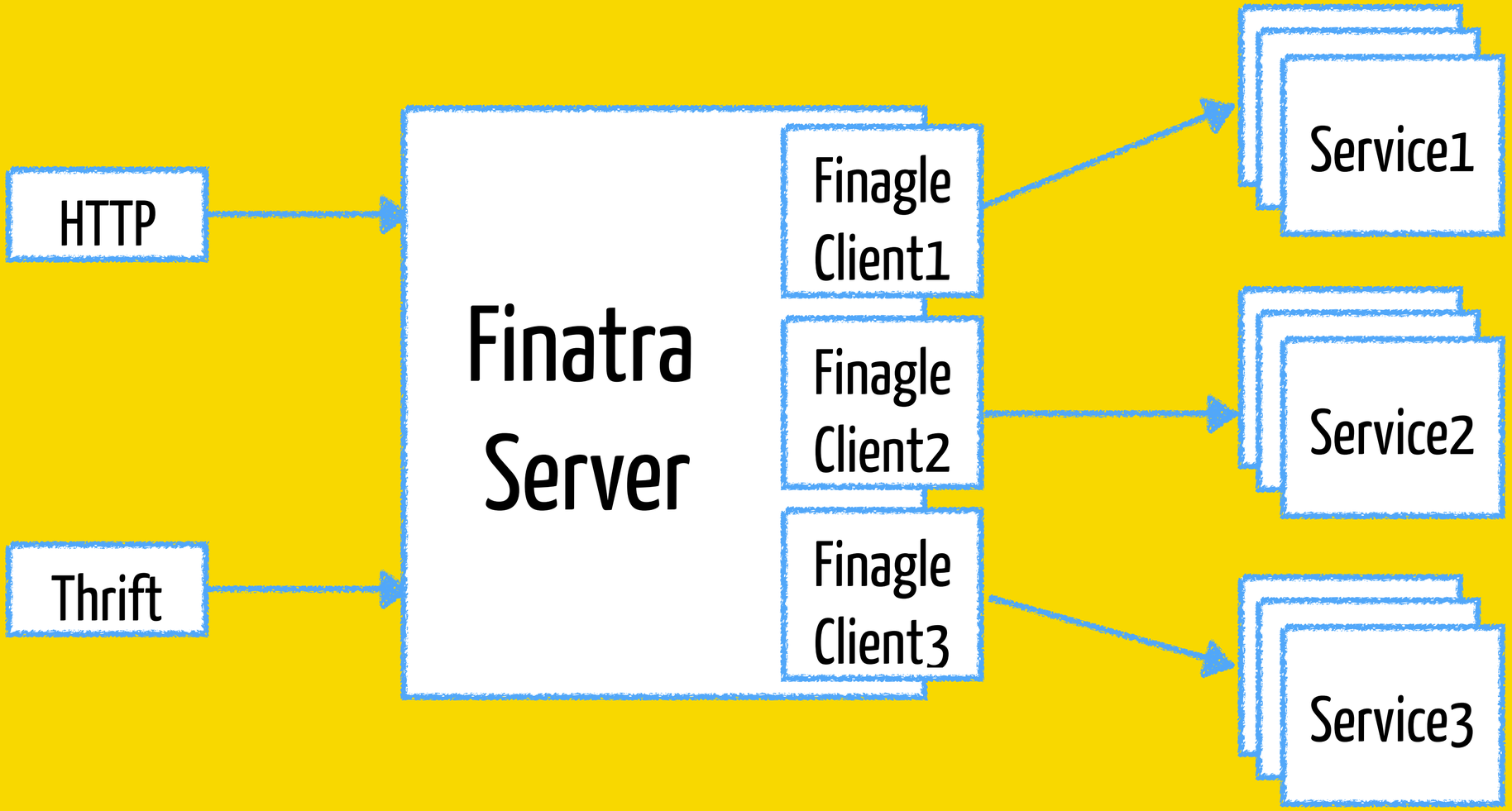
```
type ReaderToFuture[Req, Rep] = ReaderT[Future, Req, Rep]

object ReaderToFuture extends KleisliInstances {
  def apply[Req, Rep](f: Req => Future[Rep]): ReaderToFuture[Req, Rep] = Kleisli.kleisli(f)
}

implicit object FutureMonad extends Monad[Future] {
  def point[A](a: => A): Future[A] = Future.value(a)

  def bind[A, B](fa: Future[A])(f: A => Future[B]): Future[B] = fa.flatMap(f)
}
}
~
```

It's a proxy!

```
ThriftReq =>
ReaderToFuture[Ctx, ThriftRep]
```

Example Query Pipeline

MMX: Built with Finatra & Finagle

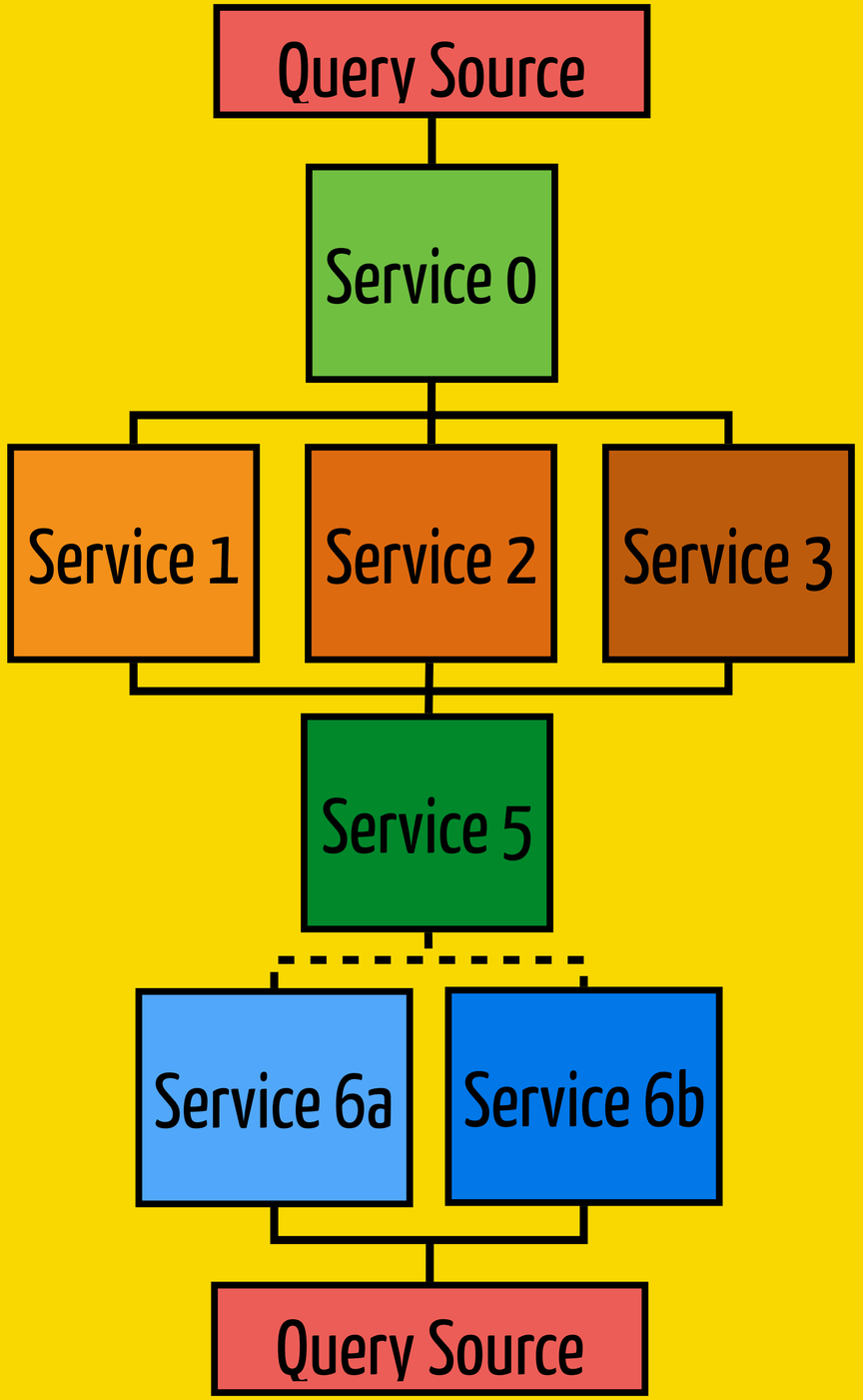


What is a Service?

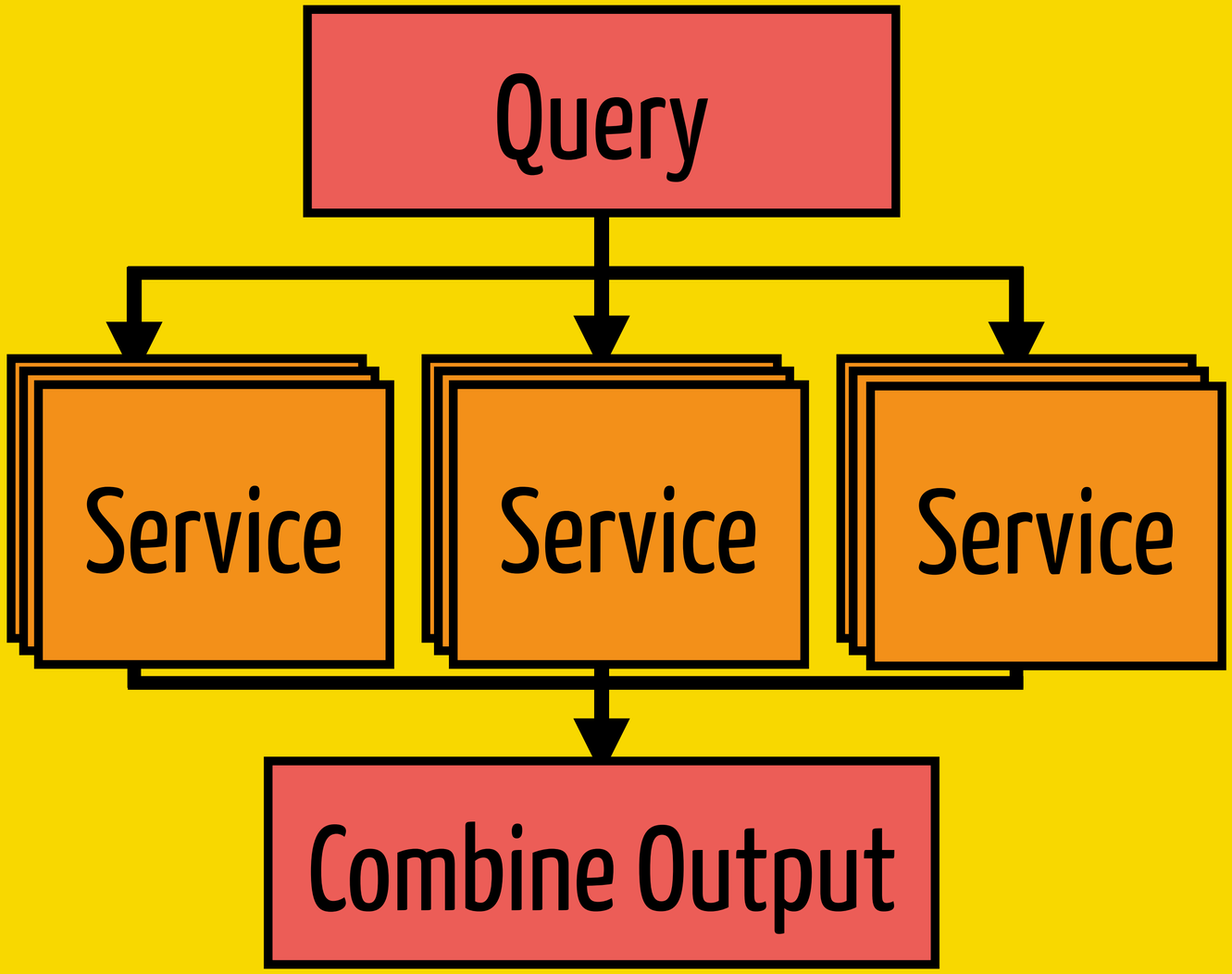
A Query Pipeline: Sequential + Concurrent Calls

call all!

call one!



Horizontally Scalable



Service building blocks

- **Search Indexes**
  - Solr
  - ElasticSearch
- **Feature & Candidate stores**
  - Arizona —> Key-Value Store
  - BigTable / DynamoDb
  - Mysql
- **Cached model output**
  - Memcache / Redis

VOWPAL WABBIT

The Incredible PYTORCH

TensorFlow

Service building blocks

- **Models** → Hosted on a separate service.
  - Engine agnostic: **Vowpal Wabbit** or **tensorflow** or **pytorch**?
  - Json Input → Output over HTTP
  - Send batch calls with features → get back scores.
    - **Finagle** : send parallel batches!
- **External APIs**
  - Any cloud provider ML APIs.

Models as a service!

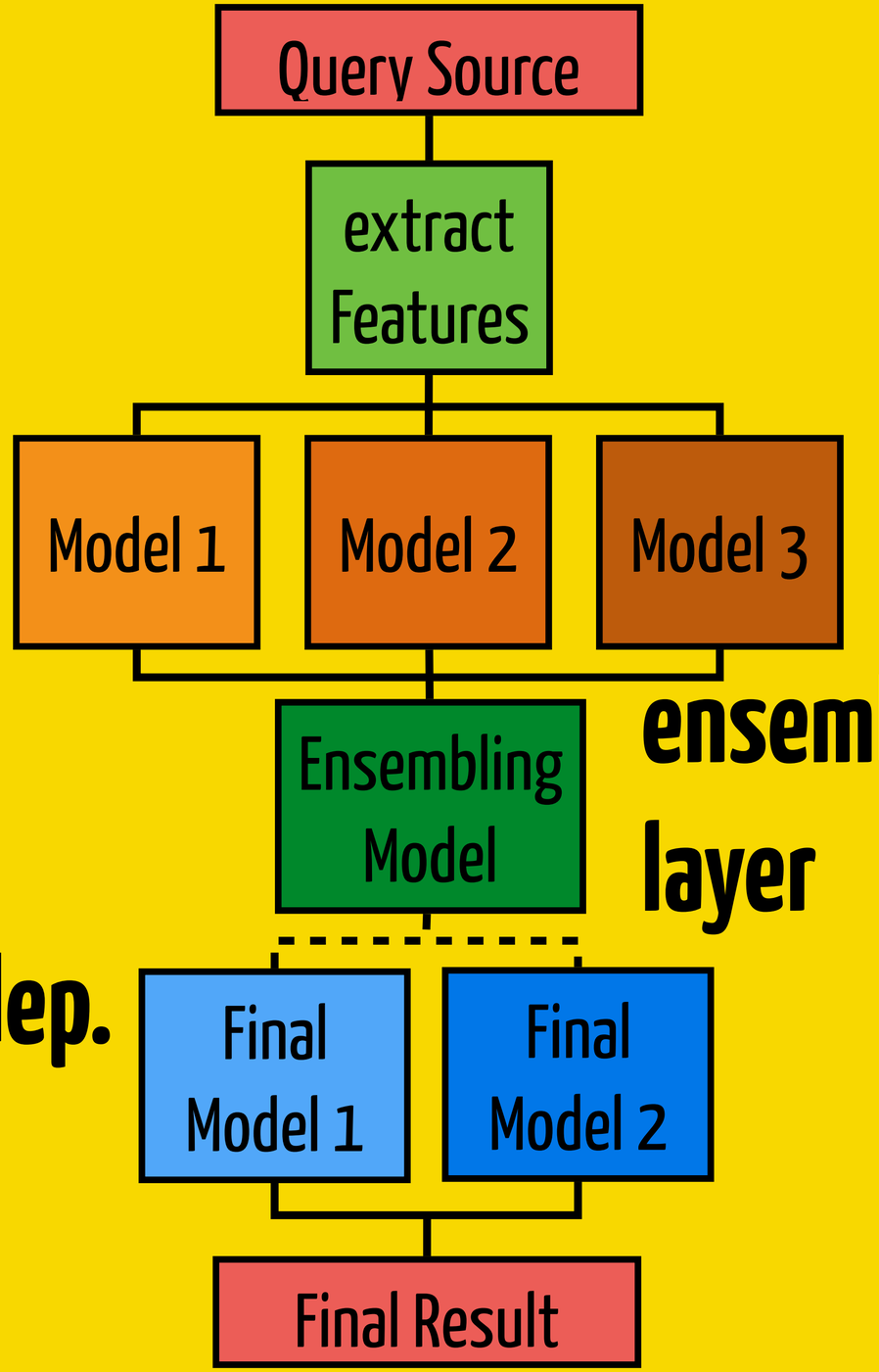
- **Inference in real time**
  - No precomputed values → No **fixed domain**
  - Separate update schedules for model & feature stores
- Each model **scales independently.**
  - Allows sub-models to be more expensive.

ENSEMBLE OVER MODELS IN REAL TIME!

- Each model gets updated independently.
- The combining function can be a model itself.
- Can define failure scenarios for partial failure.

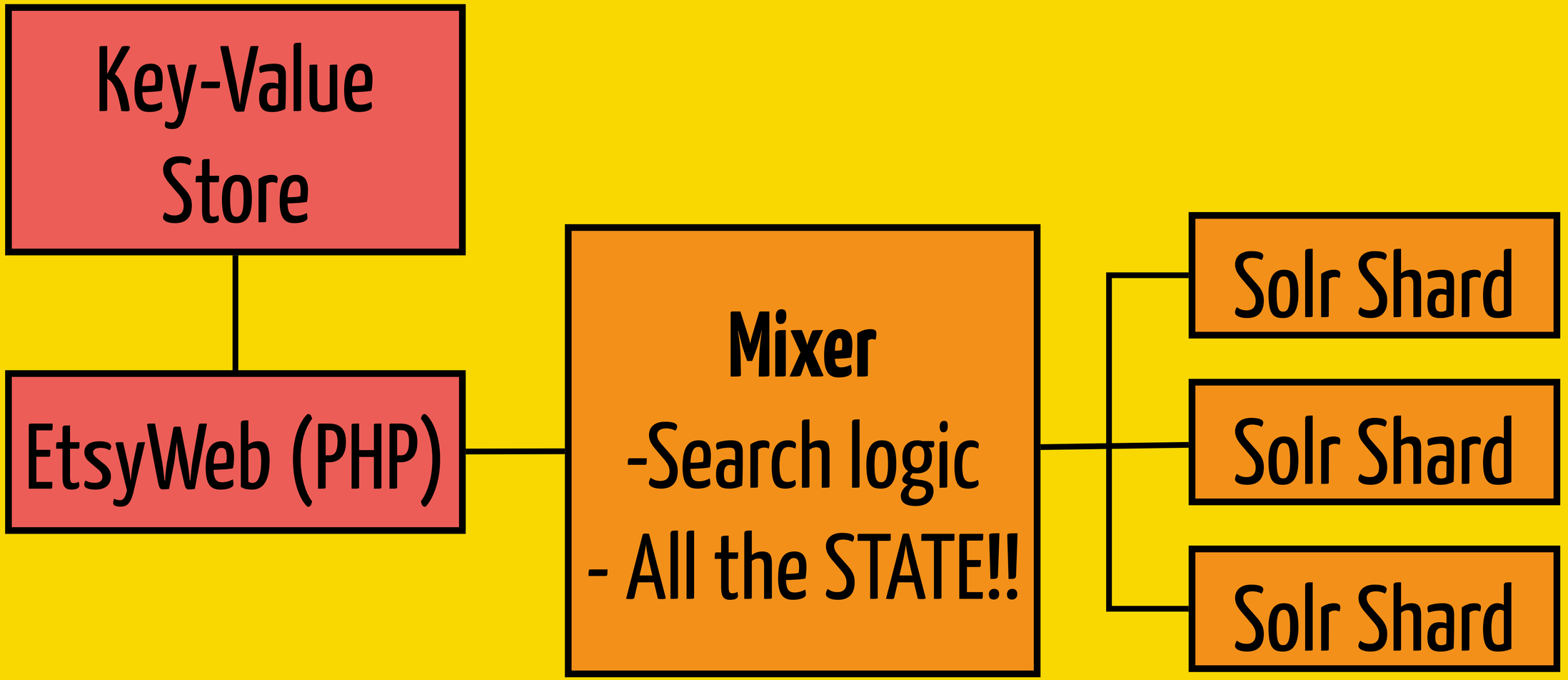
Example of Model Ensembling at Runtime.

call all in parallel!

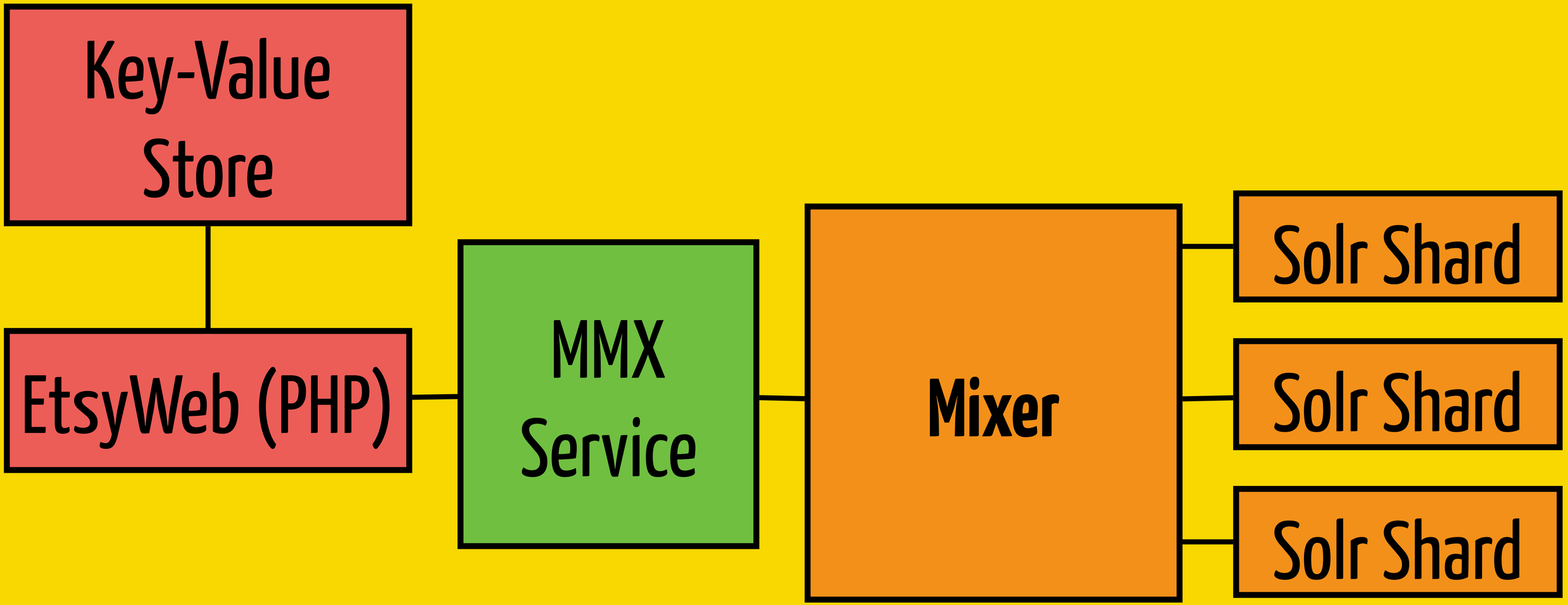


call one, dep. on score!

Search Stack before MMX



Search Stack with MMX



Before

- Java / PHP
- Models tightly coupled to search code
- Query being mutated and then “executed”
- Cost of change is high

After

- Scala / Finagle / Finatra.
- Models as a service, scale independently.
- **Real time inference pipelines.**
- Build a query plan with ReaderToFuture.
- Async communication with sub-services.

State woven into one service => Service DAG from config.

no pencils for “pencils” ʊ_ʊ

Etsy | pencils | Search | Home | Favorites | Your shop | You | Cart

Jewelry & Accessories | Clothing & Shoes | Home & Living | Wedding & Party | Toys & Entertainment | Art & Collectibles | Craft Supplies & Tools | Vintage

personalized pencils | pencils with quotes | funny pencils | grammar pencils | cute pencils | engraved pencils | mechanical pencils | pens | pencil case

All categories
- Art & Collectibles
- Craft Supplies & Tools
- Bags & Purses
- Paper & Party Supplies

+ Show more

Shipping
- Free shipping
- Ready to ship in 1 business day
- Ready to ship within 3 business days

Special offers
- On sale

Color
- Black
- White
- Blue
- Pink
- Green

+ Show more

Shop location
- Anywhere
- United States
- Custom

Enter location

Item type
- All items
- Handmade
- Vintage

Price ($)
- Any price
- Under $25
- $25 to $50
- $50 to $100
- Over $100

All categories › “pencils” (244,159 Results)

Sort by: Relevancy

More colors

Julius Caesar Pencil Holder (with Cust...
ElevatedInnovations
★★★★★ (2)
$15.00

Teacher SVG, DXF File, Beaching Not T...
TheSmudgeFactoryLLC
★★★★★ (3,850)
$3.99

Original Colored Pencil Drawing, Red ...
ArtCornerShop
★★★★★ (239)
$100.00

Rise Up Pencil Set
ThePoshShopCo
★★★★★ (176)
$10.00

Small African Print Pouch, Ankara Smal...
LollyKiks
★★★★★ (48)
$20.78

A5 storage tin with hinged lid | Simple,...
WhirlingWorld
★★★★★ (84)
$5.54

Miss Doughmestic Thank You with Pen...
BobbisCookiesCutters
★★★★★ (9,539)
$6.00

Elephant wall art print / Pencil / digital ...
BringGoodMood
(9)
$3.90

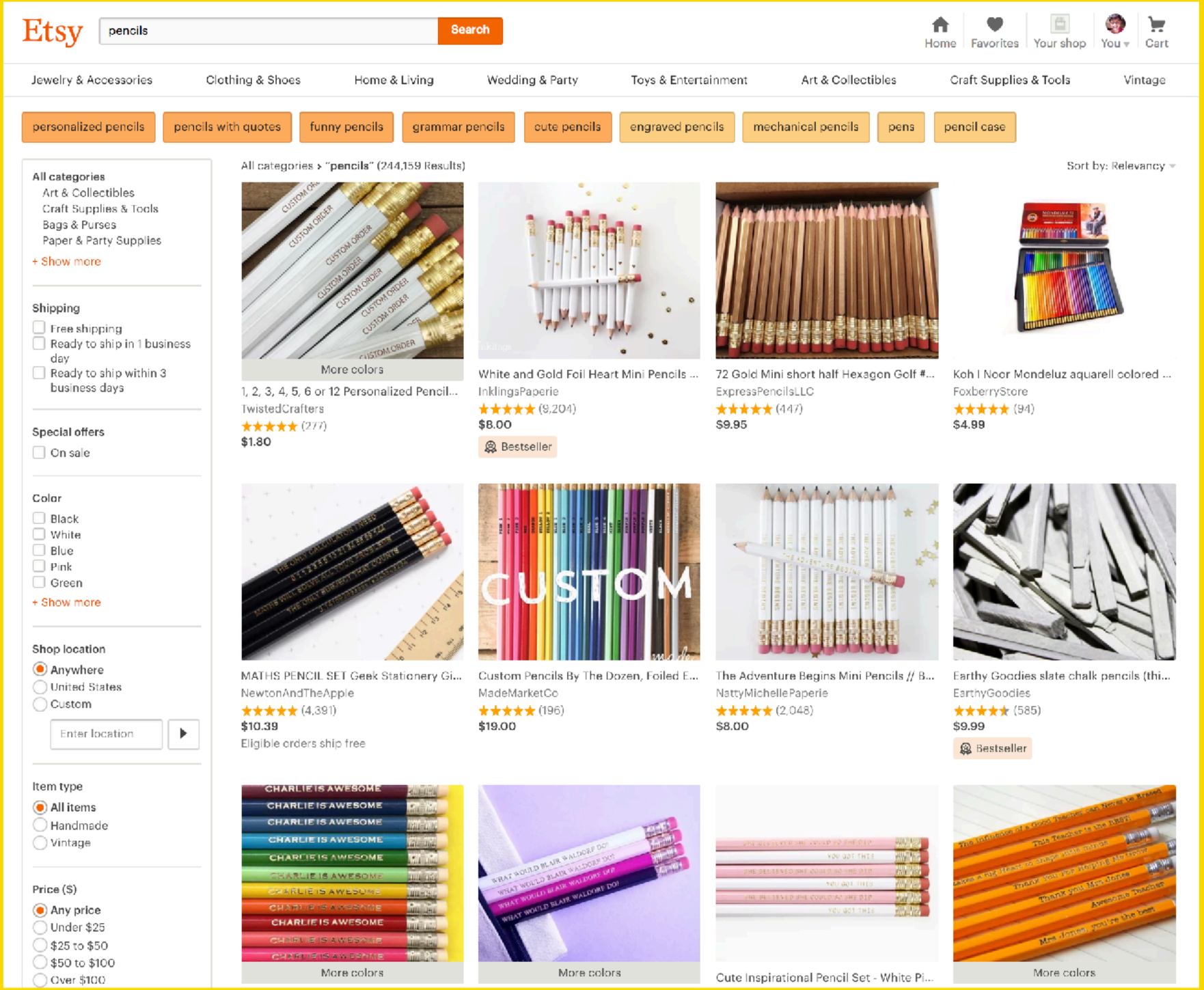
Framed Art: original 8x10 inch pencil d...
ARTuro1984
$59.99 Free shipping

Thanos Portrait - Colored Pencil Drawi...
JimmiePetrascuART
$19.99

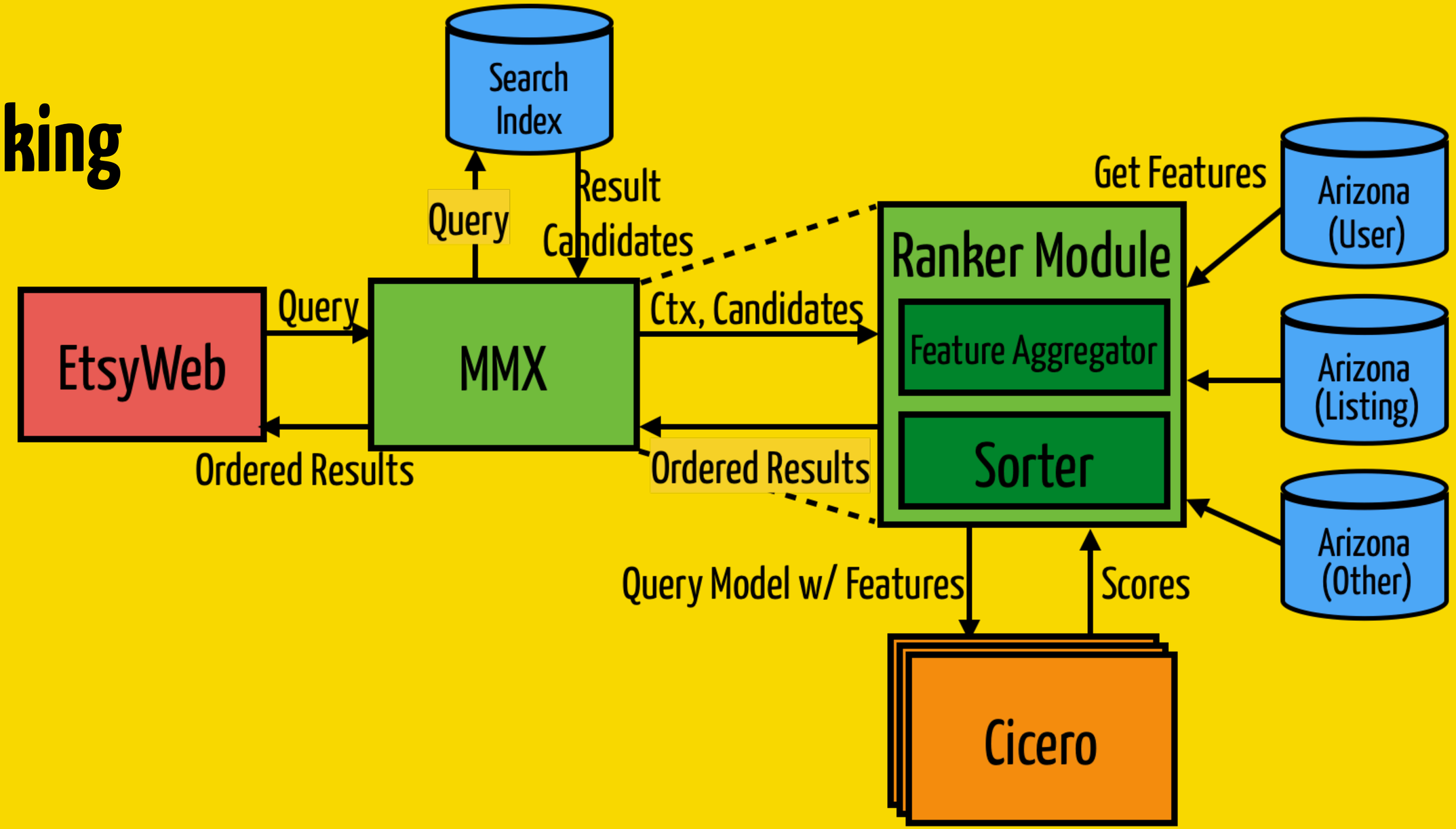
vtg 50s flannel pencil skirt high waist ...
mad4modvintage
★★★★★ (375)

Color Pencil Rose
StitchMadnessMarket
$8.00 Free shipping

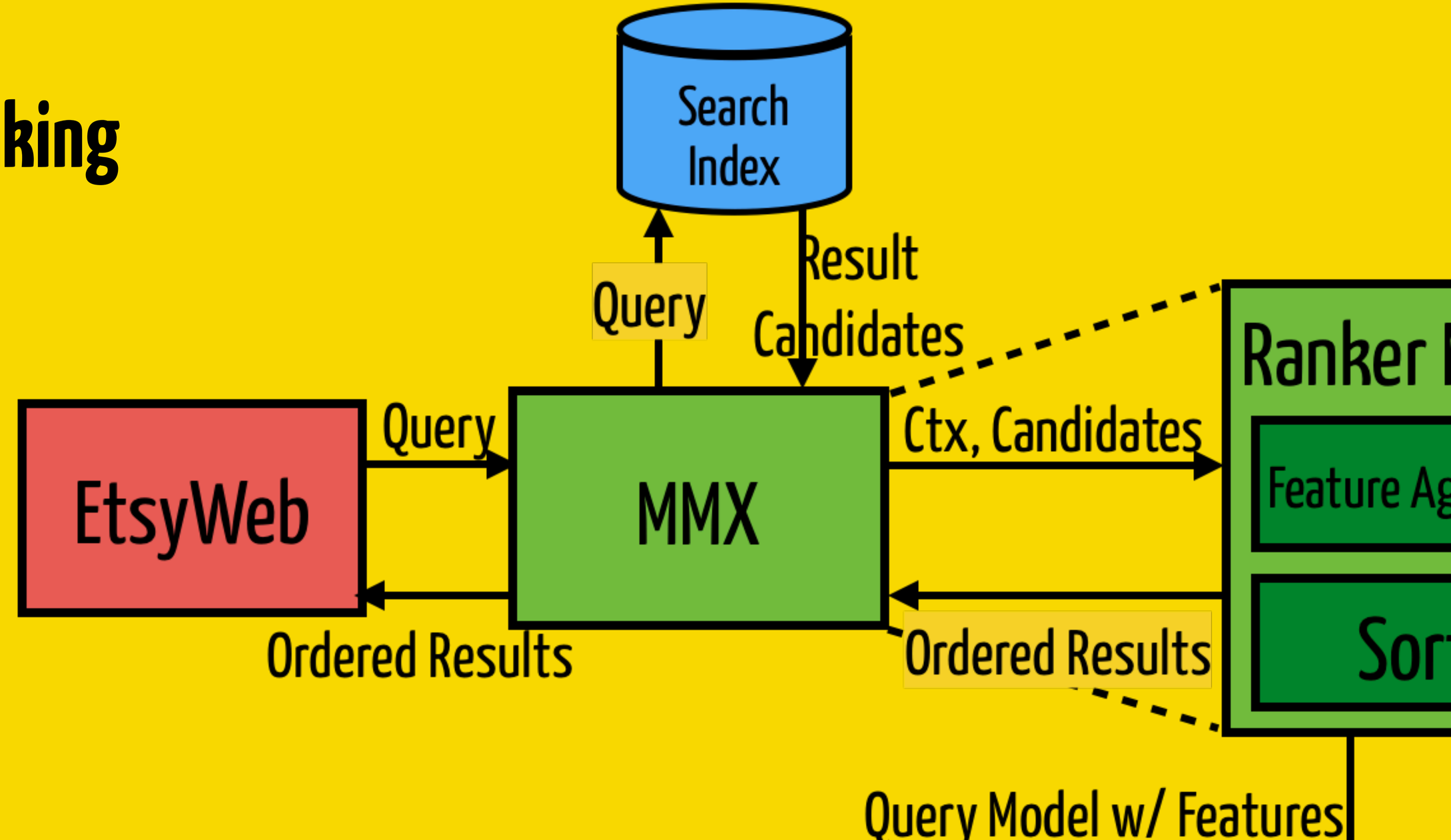
MMX + Query Ranking Model

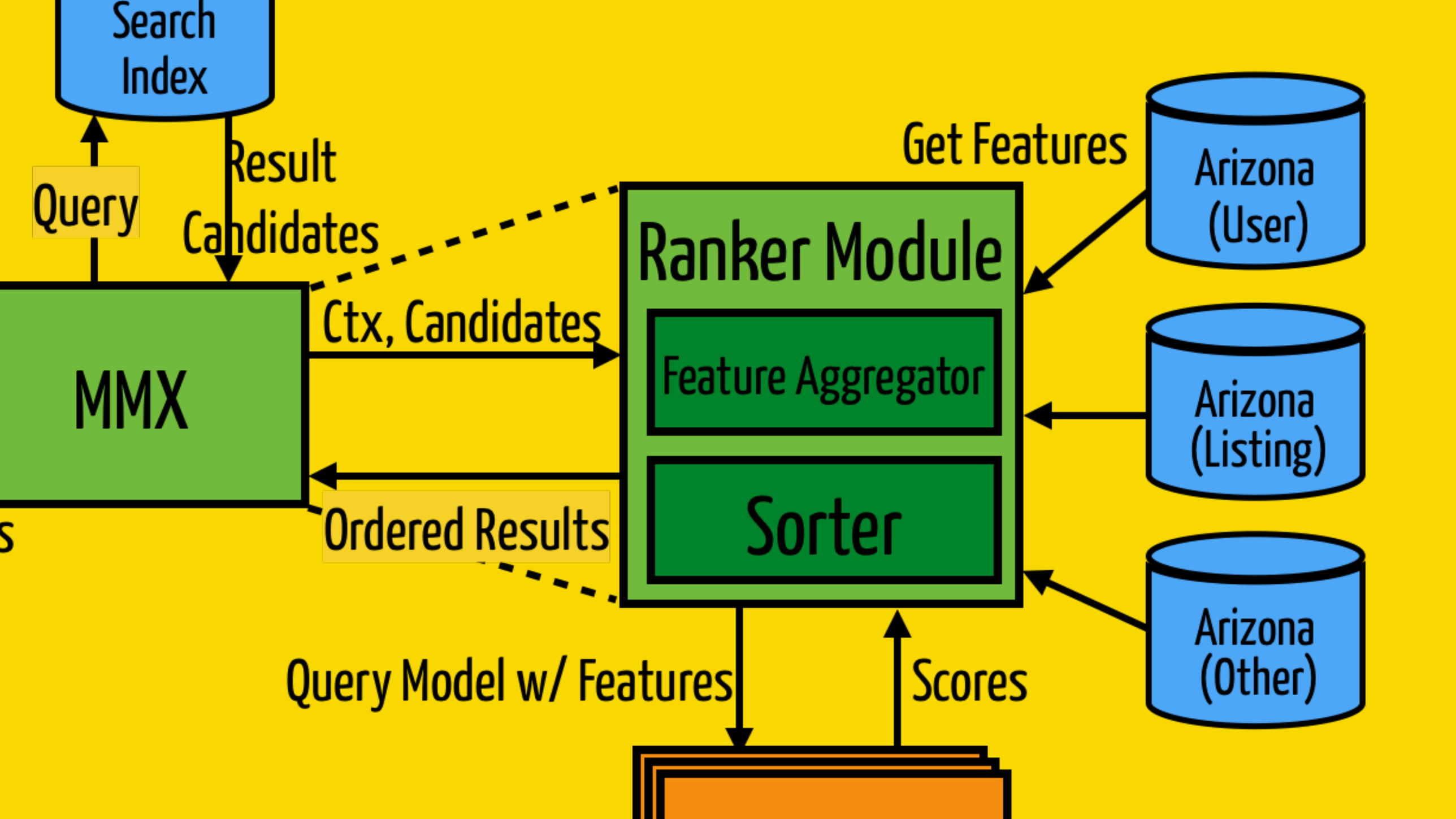


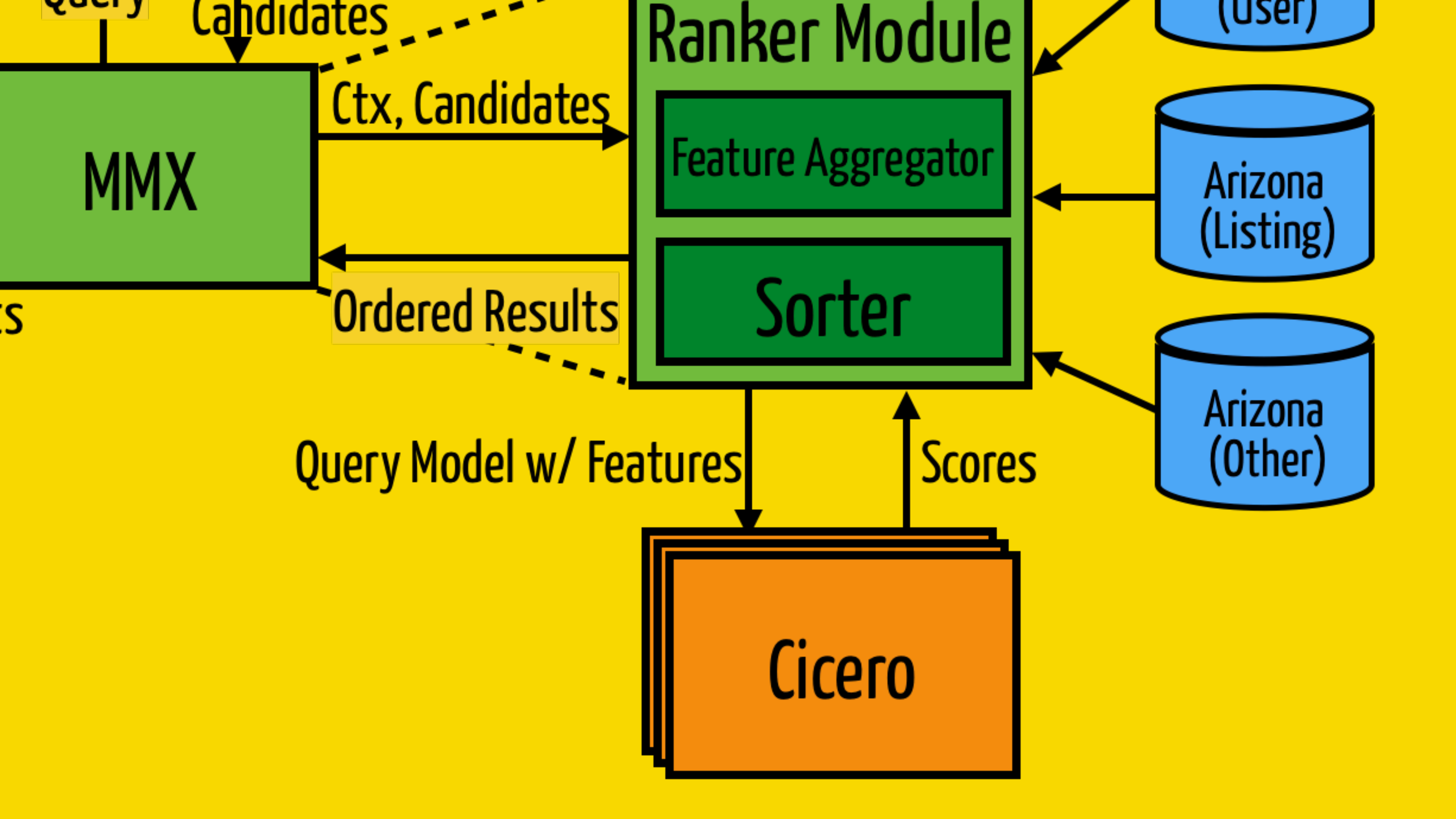
New Ranking Pipeline

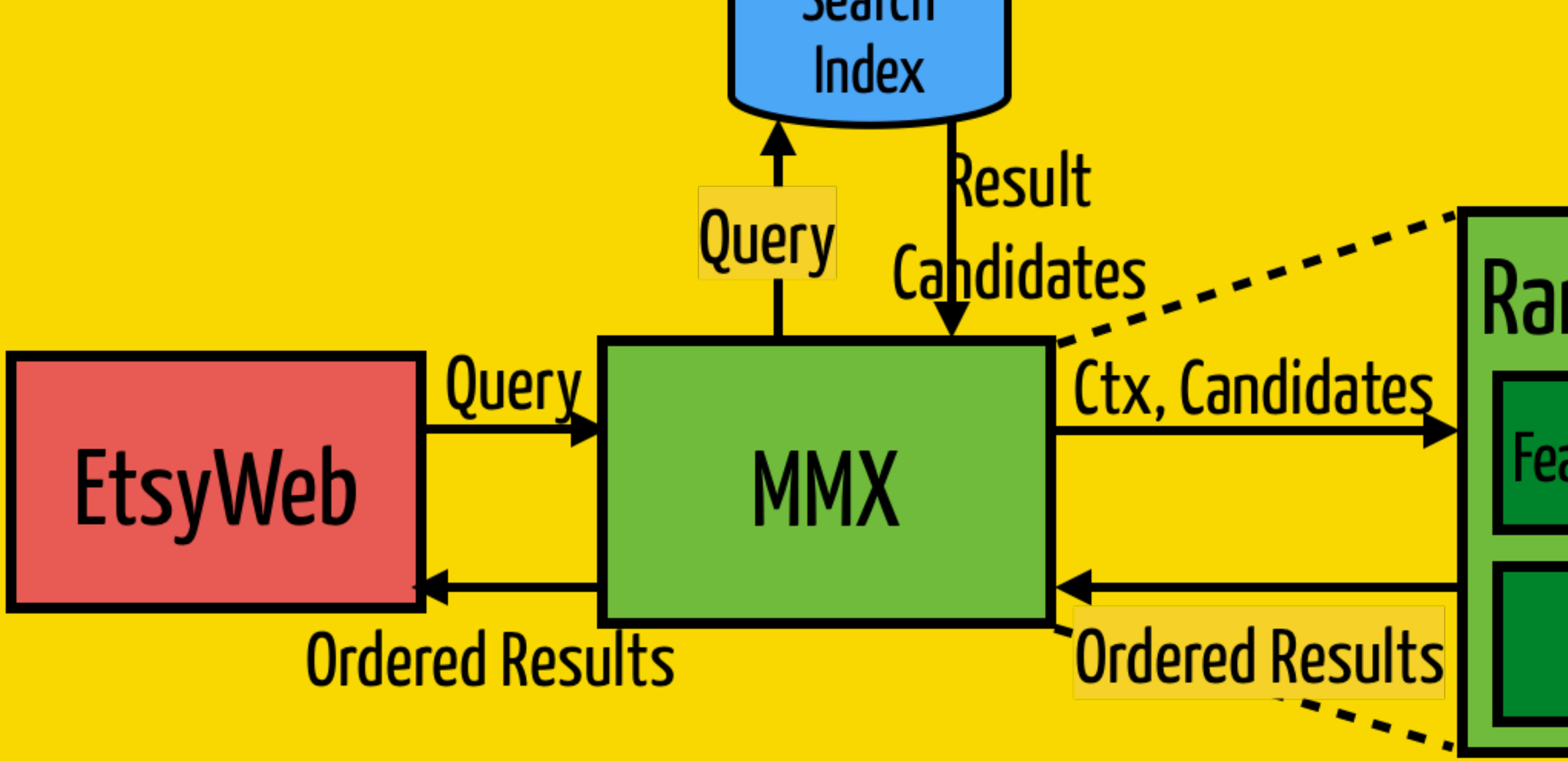


New Ranking Pipeline



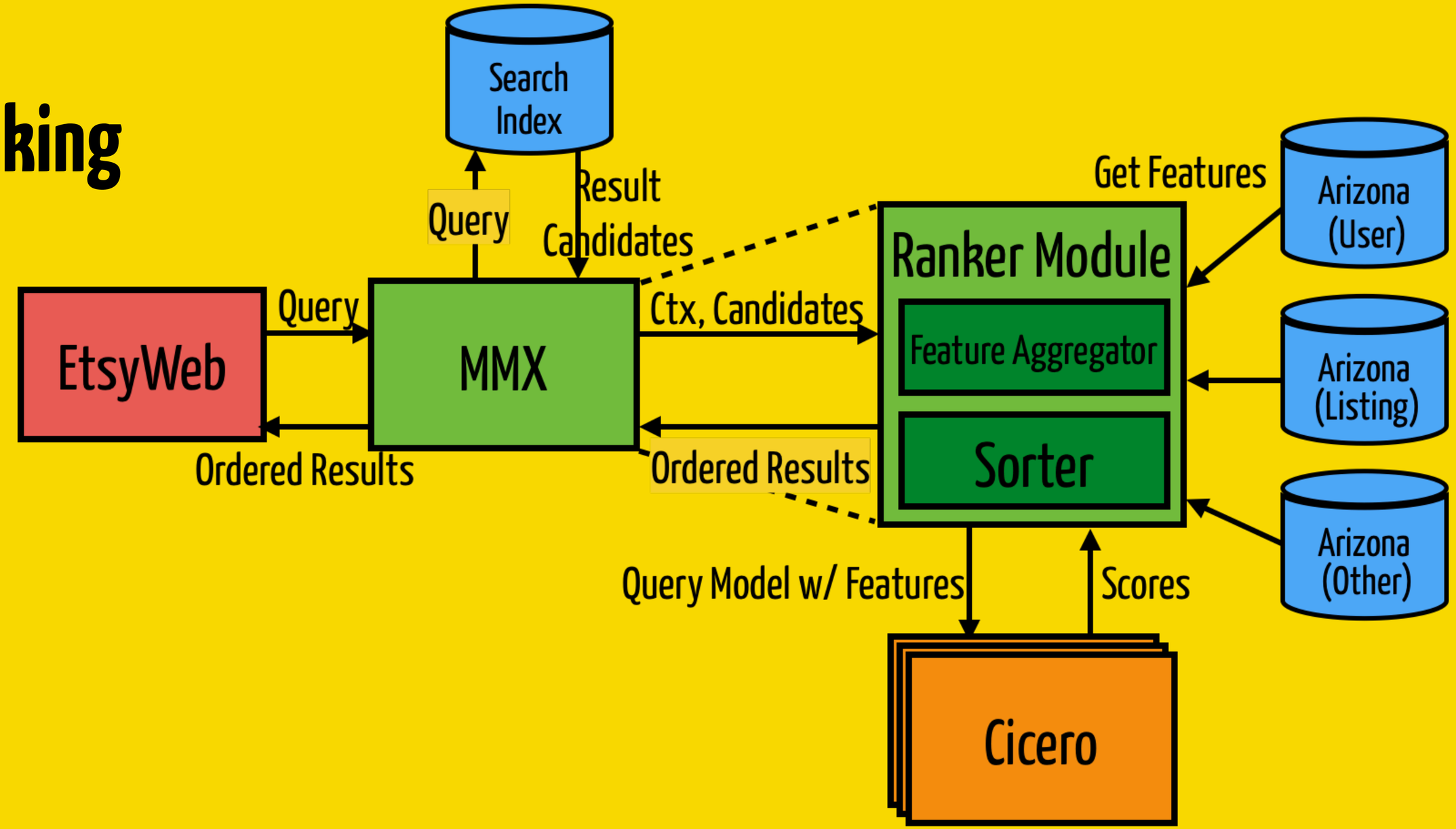

Search Index
Query
Result Candidates
Get Features
Arizona (User)
Ranker Module
Feature Aggregator
Sorter
Ctx, Candidates
MMX
Arizona (Listing)
Ordered Results
Arizona (Other)
Query Model w/ Features
Scores


Candidates
Ranker Module
(User)
MMX
Ctx, Candidates
Feature Aggregator
Arizona
(Listing)
Ordered Results
Sorter
Arizona
(Other)
Query Model w/ Features
Scores
Cicero



Query Model w/ Feature

New Ranking Pipeline



No Results ⇒

Query Backoff

Query Backoff

TV 1340ms + 211ms + 4676ms = 6227ms

Etsy | kayne west sneakers for puppies | Search | Home | Favorites | Your shop | You | Cart

Jewelry & Accessories | Clothing & Shoes | Home & Living | Wedding & Party | Toys & Entertainment | Art & Collectibles | Craft Supplies & Tools | Vintage

Special offers
- On sale

All categories
- Shoes
- Art & Collectibles
- Pet Supplies
- Toys & Games

+ Show more

Shipping
- Free shipping
- Ready to ship in 1 business day
- Ready to ship within 3 business days

Shop location
- Anywhere
- United States
- Custom

Enter location

Item type
- All items
- Handmade
- Vintage

Price ($)
- Any price
- Under $25
- $25 to $50
- $50 to $100
- Over $100
- Custom

Showing results for **sneakers for puppies.**

kayne west sneakers for puppies had no results.

Search instead: kayne west sneakers | west sneakers | sneakers

All categories › "sneakers for puppies" (81 Results)

Sort by: Relevancy

Fabric, Dawgs and Sneakers on Brown, Funny ...
BluePacificFabrics
$10.00
★★★★★ (2,446)

Custom Personalize Design Your Waterproof P...
DGCustomGraphics
$39.00
★★★★½ (114)

It's A Ruff Life Nike Roshe Run Black Custom P...
NYCustoms
$135.00 ~~$150.00~~ (10% off)
★★★★½ (1,211)

Sneakers the Corgi Pillow (Mermaid)
thousandskies
$35.00
★★★★★ (1,033)

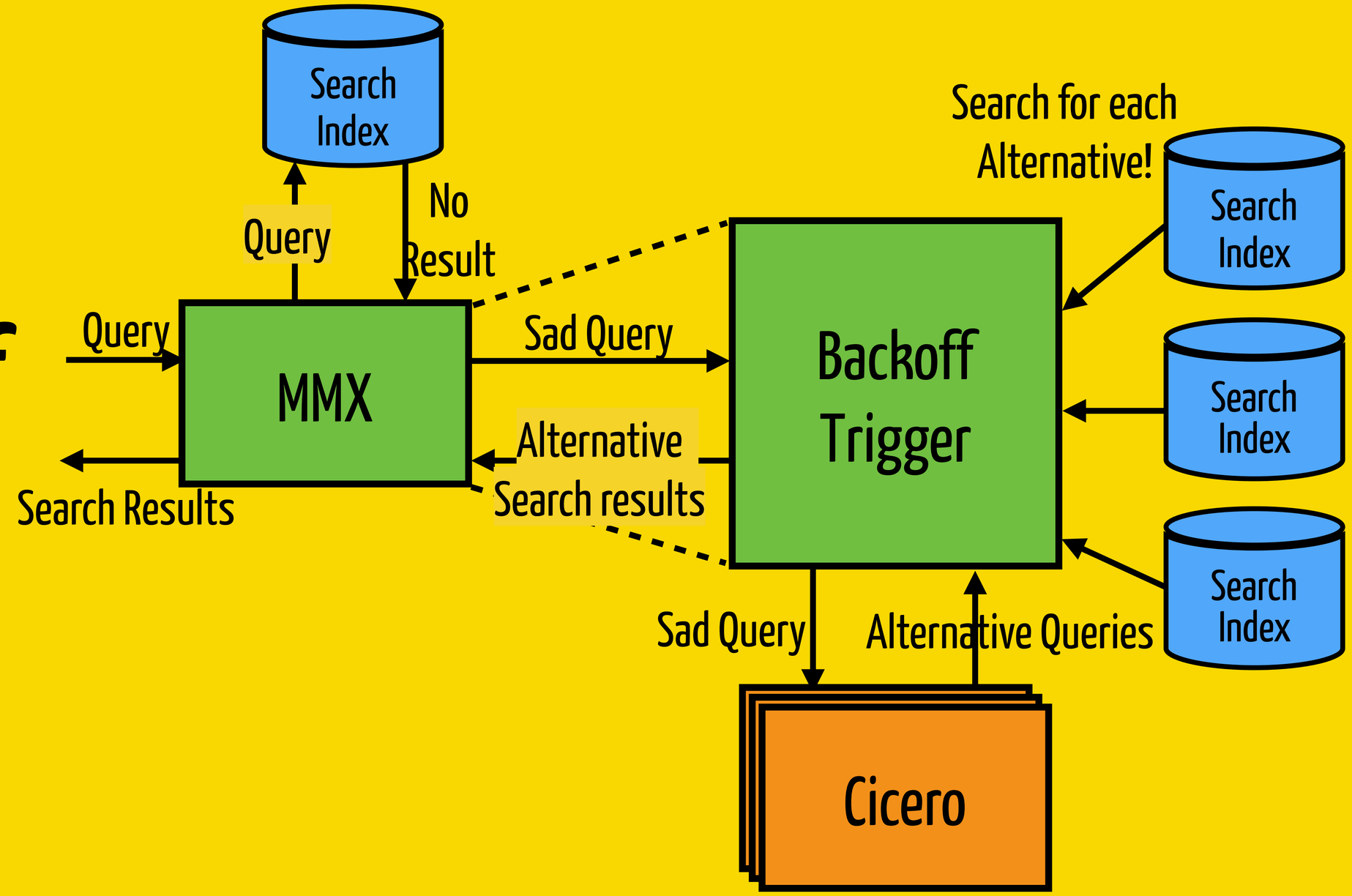
Naughty Dachshund - Women's Sneakers with ...
CustomKIKScom
$69.95

Sneakers the Corgi Pizza Pillow (Teal collar)
thousandskies
$30.00

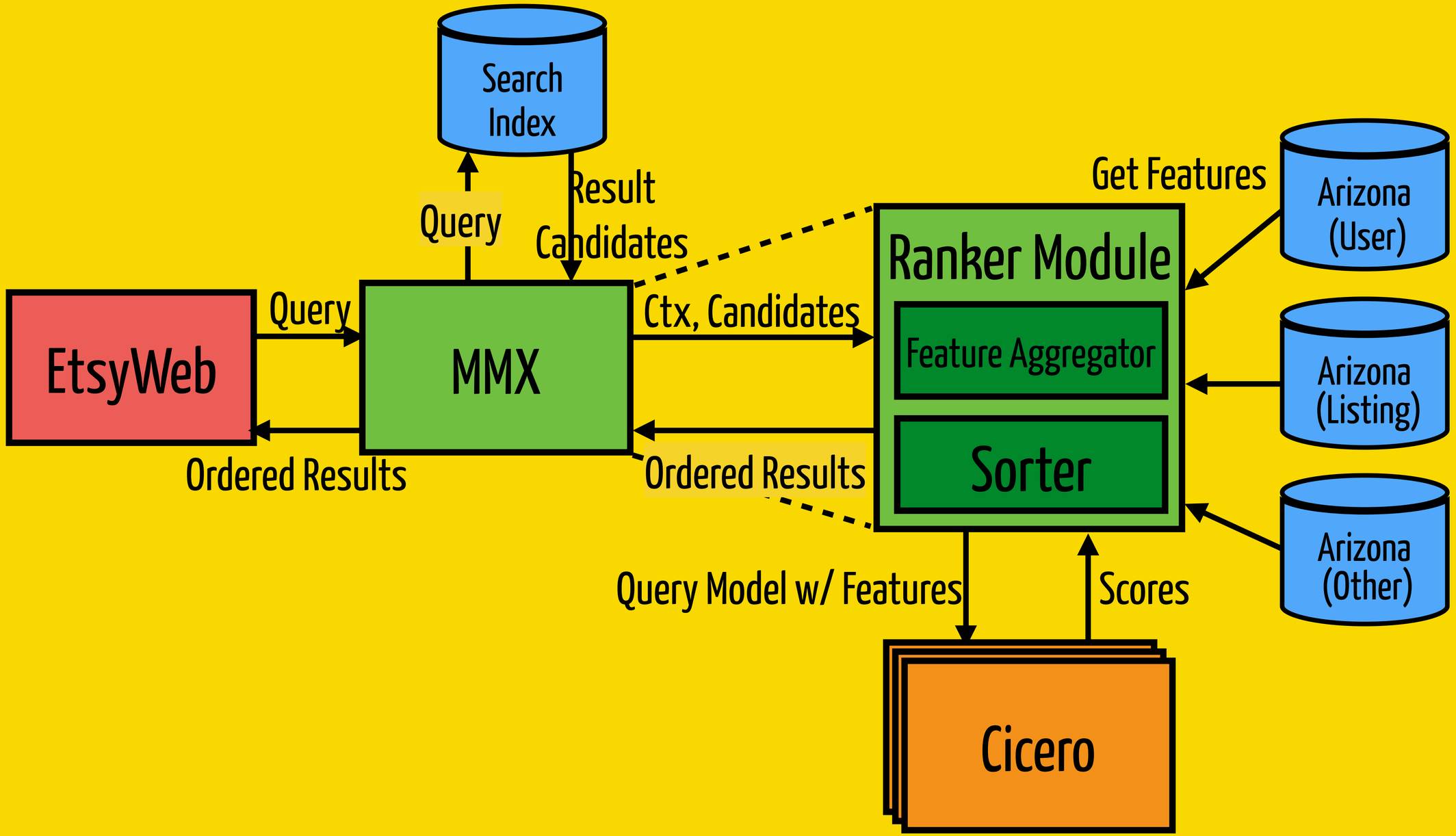
vintage womens Hush Puppies golf shoes - ta...
humblepeacock
$75.00

Girls High Top Puppy Shoes Custom High Top...
PricklyPaw
$78.50

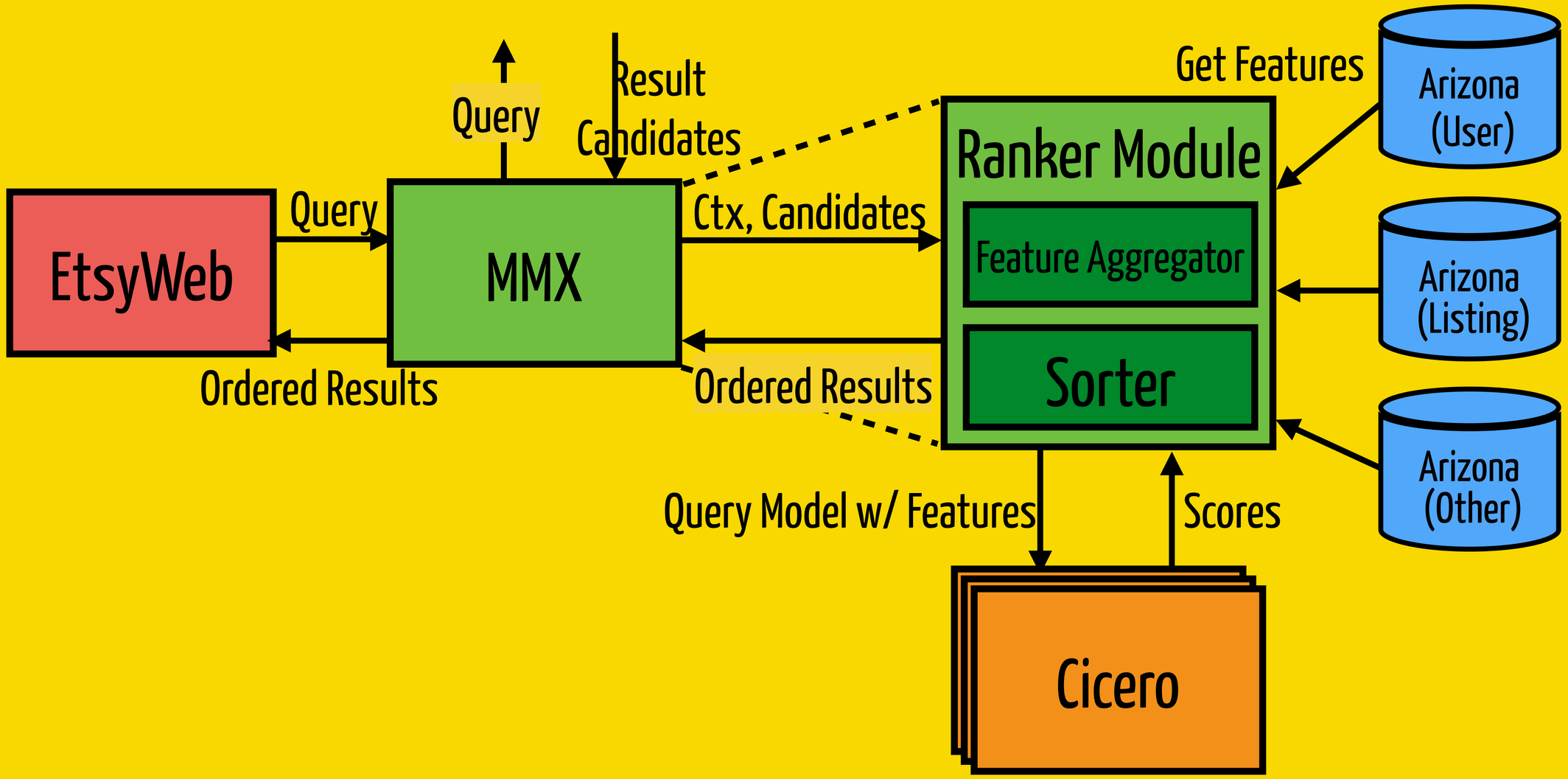
Query Backoff Pipeline



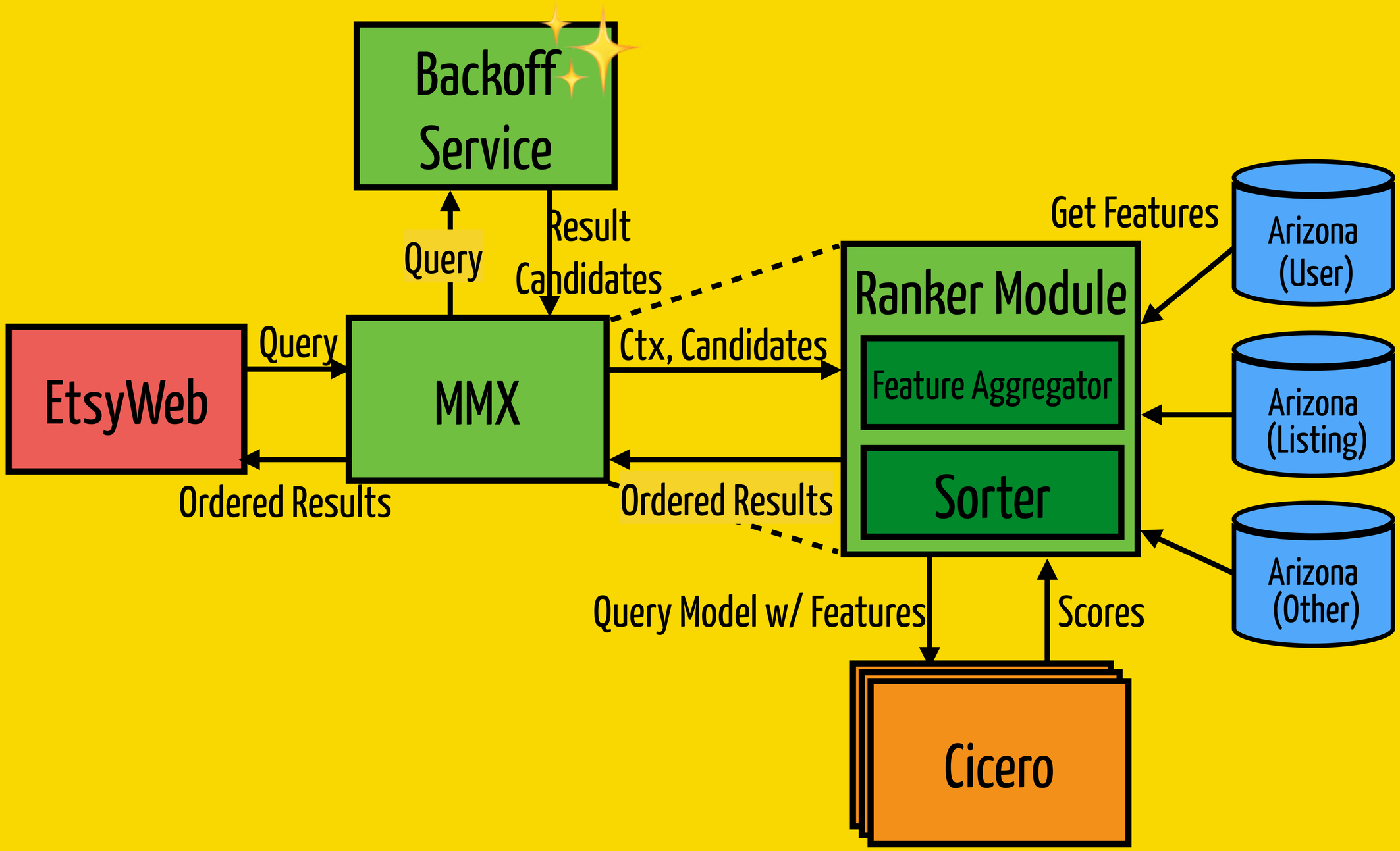
Current Ranking Pipeline



Ranking + Backoff

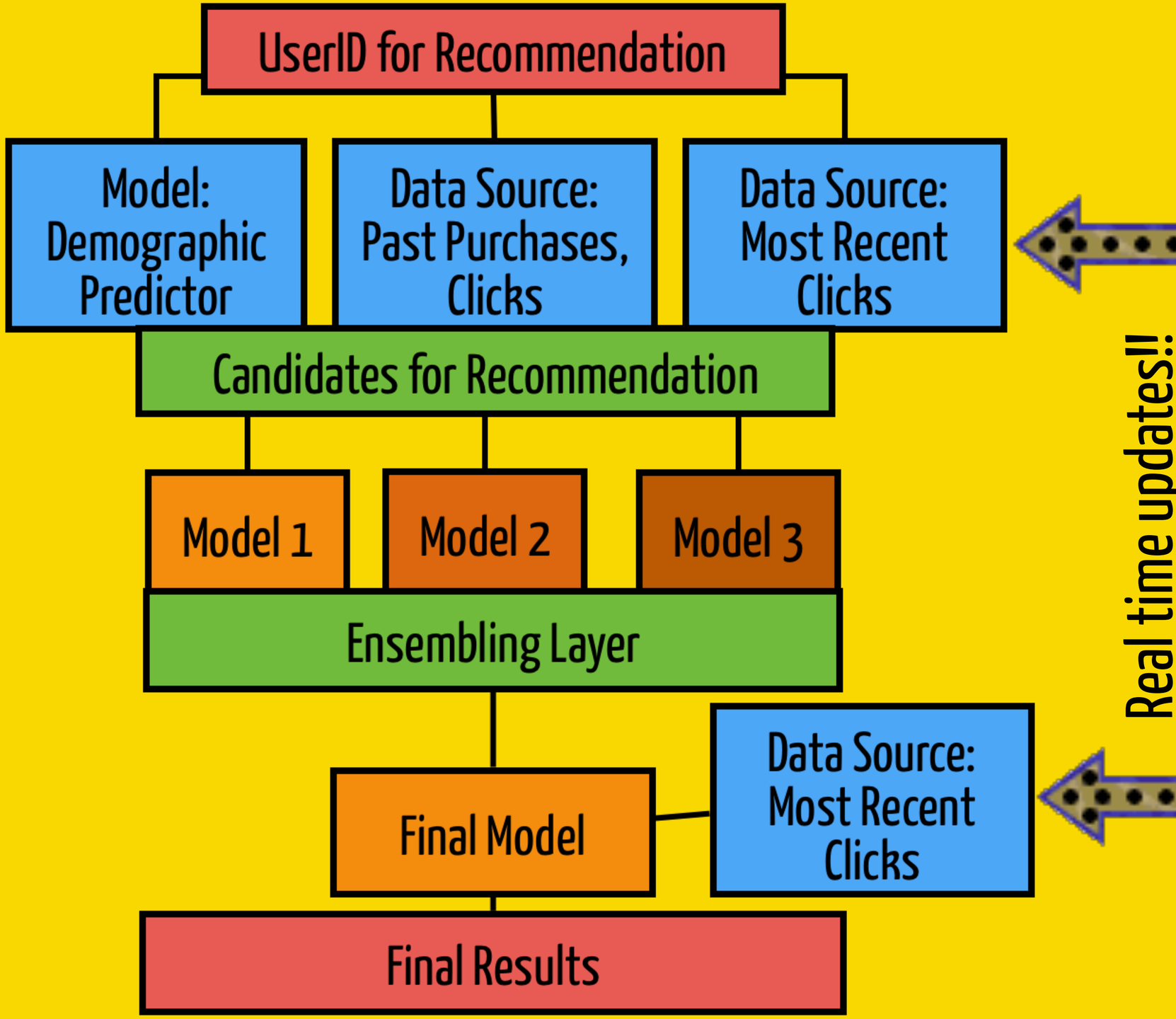


Ranking + Backoff

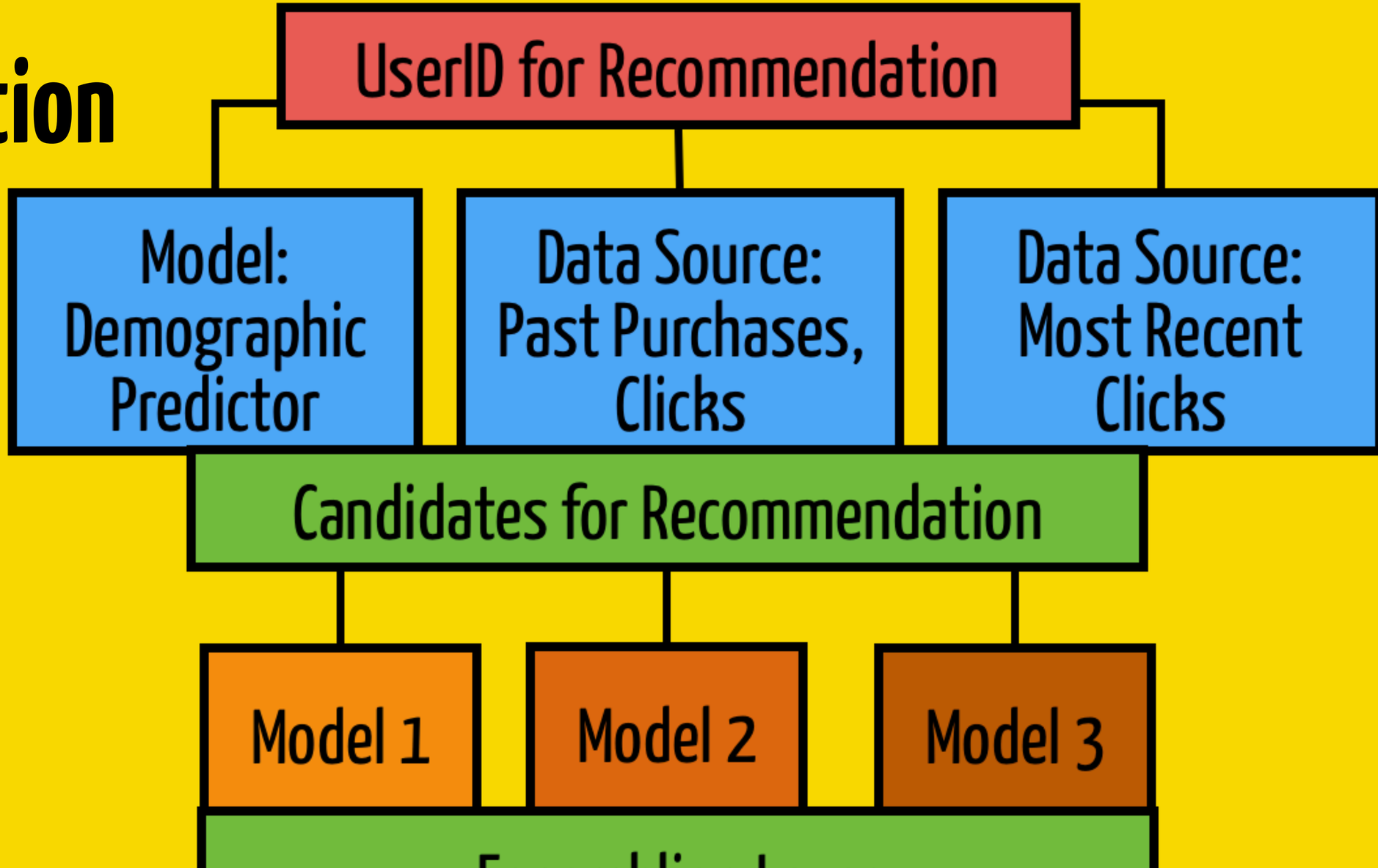


Recommendation Pipeline

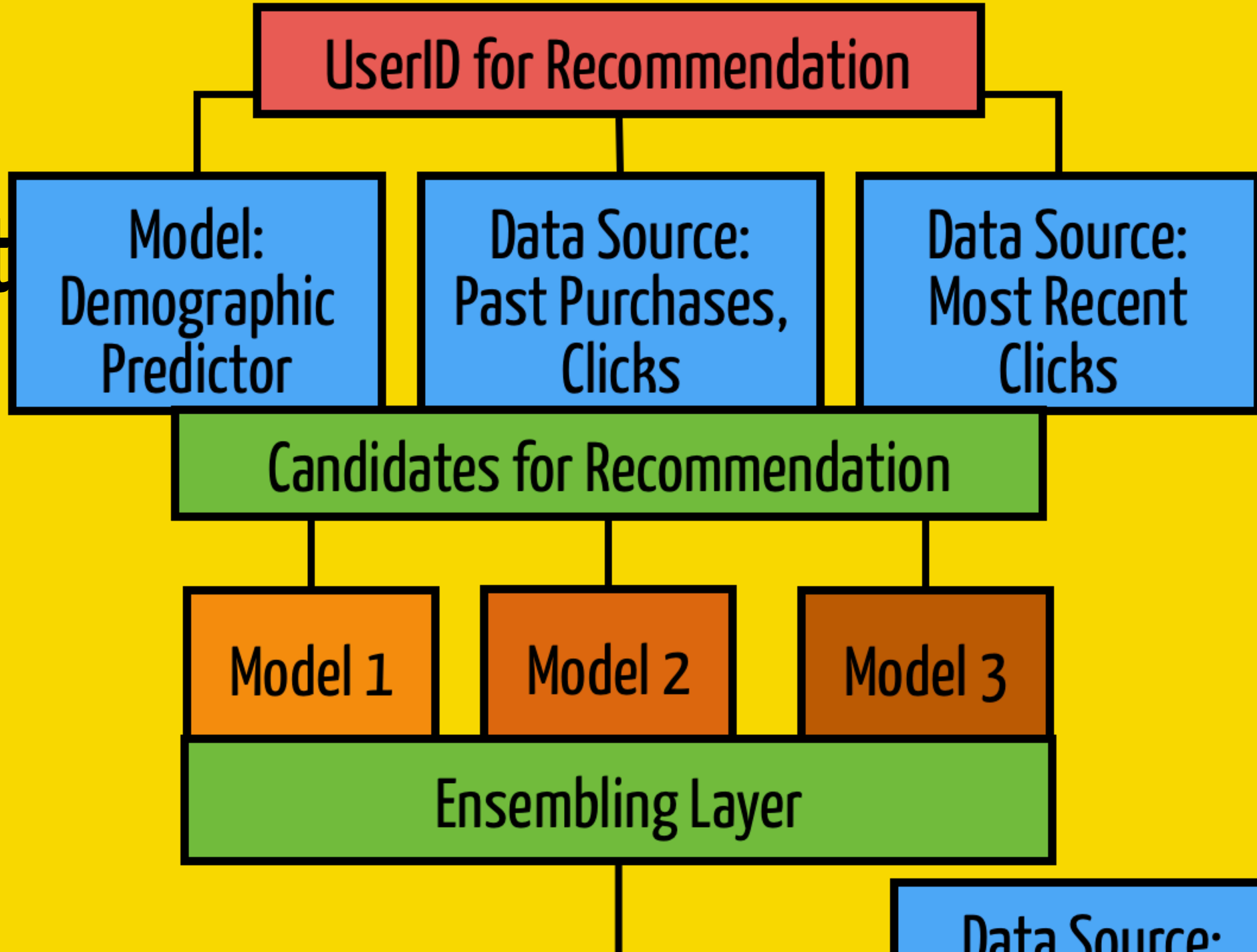
Recommendation Pipeline



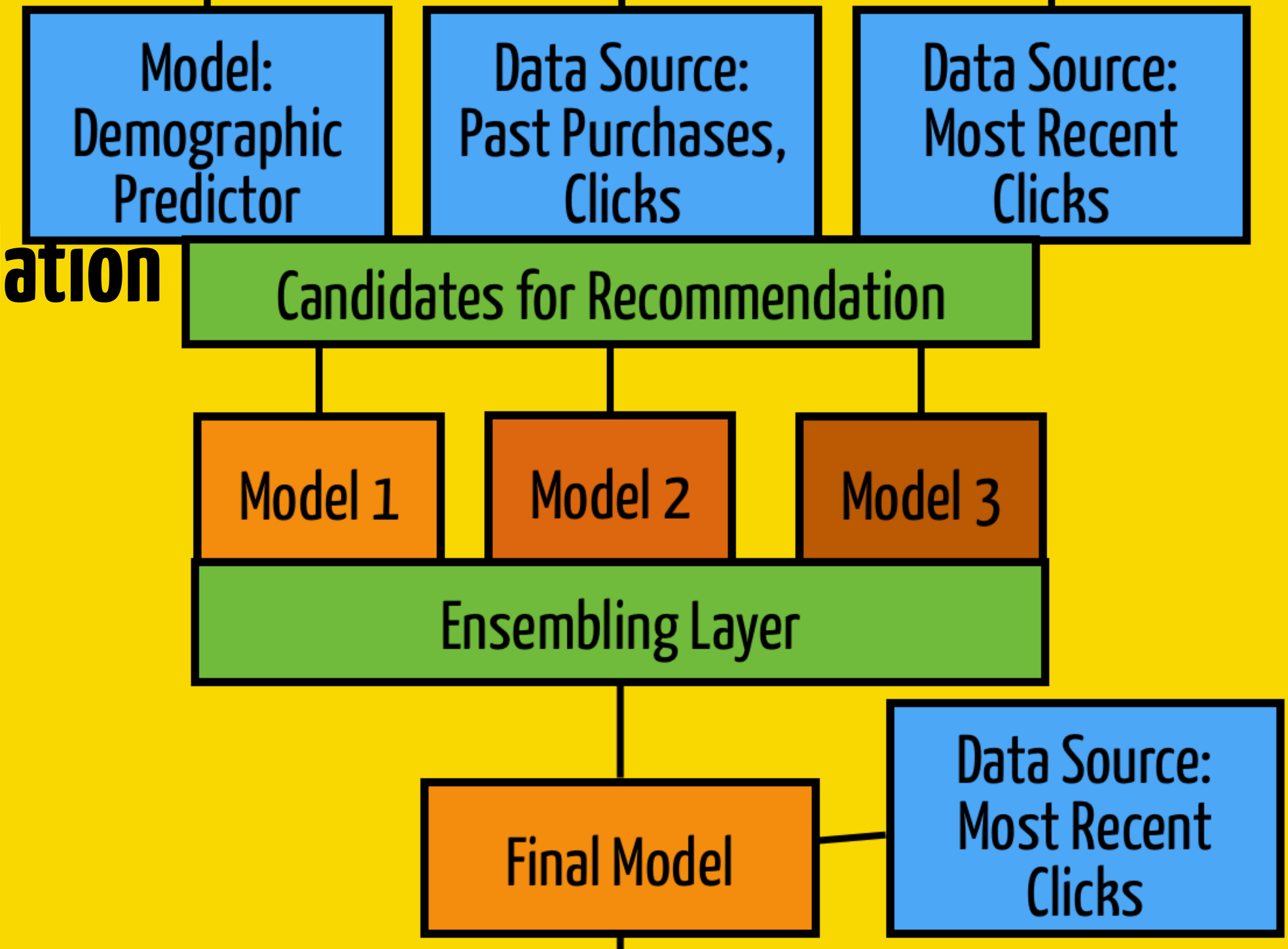
Recommendation Pipeline



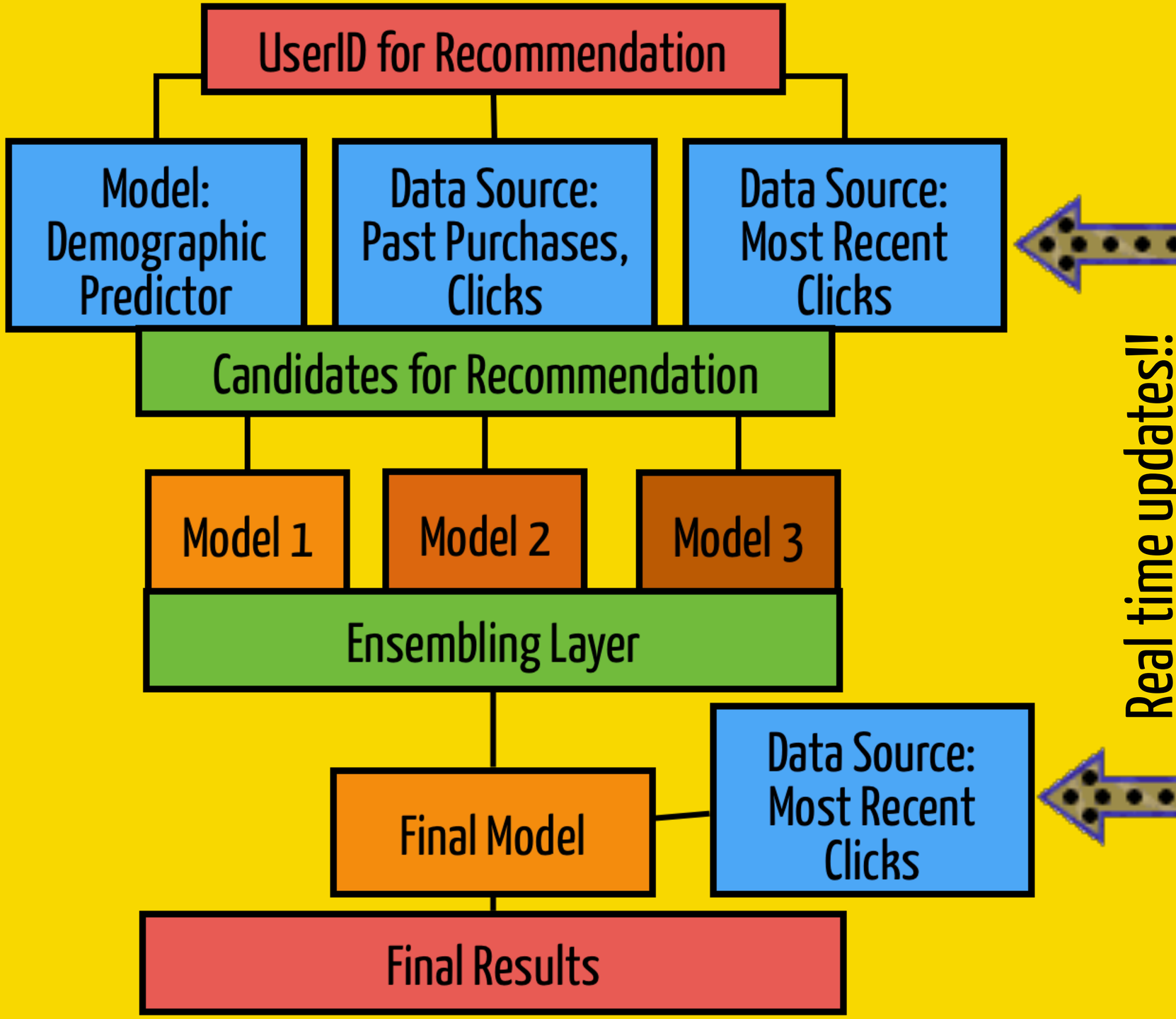
Recommendat Pipeline



Recommendation Pipeline



Recommendation Pipeline



A/B Testing

- Configurable pipelines ⇒ Create **versions of a pipeline**.
  - different model versions.
  - different arguments to services.
    - Example: Increasing reranking coverage was only a config knob switch.
  - different data sources → **Features**, **Candidates** etc
    - Pick and choose datasets, versions

A/B Testing: Configuring Variants

```
"variant_200" :  ${configs.control} {
  ranker_options: { "rank_limit": 200}
},
"variant_400" :  ${configs.control} {
  ranker_options: { "rank_limit": 400}
},
"variant_tuned_boosts" : ${configs.control} {
  ranker_options: {
      "rank_limit"   : 400
      "variant_mean" : 0.71,
      "variant_std"  : 0.53,
      "control_mean" : 0.20,
      "control_std"  : 0.95,
      "relevance"    : false,
      "missing_val"  : 0.23
  }
}
```

Takeaways

- Each service is a **reusable unit** in the query DAG.

  => **algebra over services**.
- Models as a Service => **Real time inference**.
- Building new ML query pipelines is **fun!!** 😊

  + we can debug and **experiment**.
- Towards **query understanding**!

Thank you! Danke!

Questions?

We're Stefanie Schirmer..
sschirme@gmail.com
@linse on twitter

..and Aakash Sabharwal
asabharwal@etsy.com
@aakashsabharwal on twitter