Restoring Relevance with RESO's Data Dictionary, API and Event Catalog

Mark Lesswing

Boring, but necessary slide

- I am not talking about RETS (1.5, 1.7.2 or 1.8)
- RETS is oriented towards "bulk" activities
- RESO Data Dictionary can be used in other ways
- RETS is too "hard"; developers are lazy
- DMQL too "heavy" for mobile use cases

B-School Slide

- Improves TTM by reducing complexity
- Improves efficiency of delivery
- Opens up new markets
- Increases development resource pool
- Improves customer satisfaction

Trio Courtesy of RESO

Data
Dictionary

API
Transport

Event
Catalog

What is being moved

When moving is needed

How it is being moved
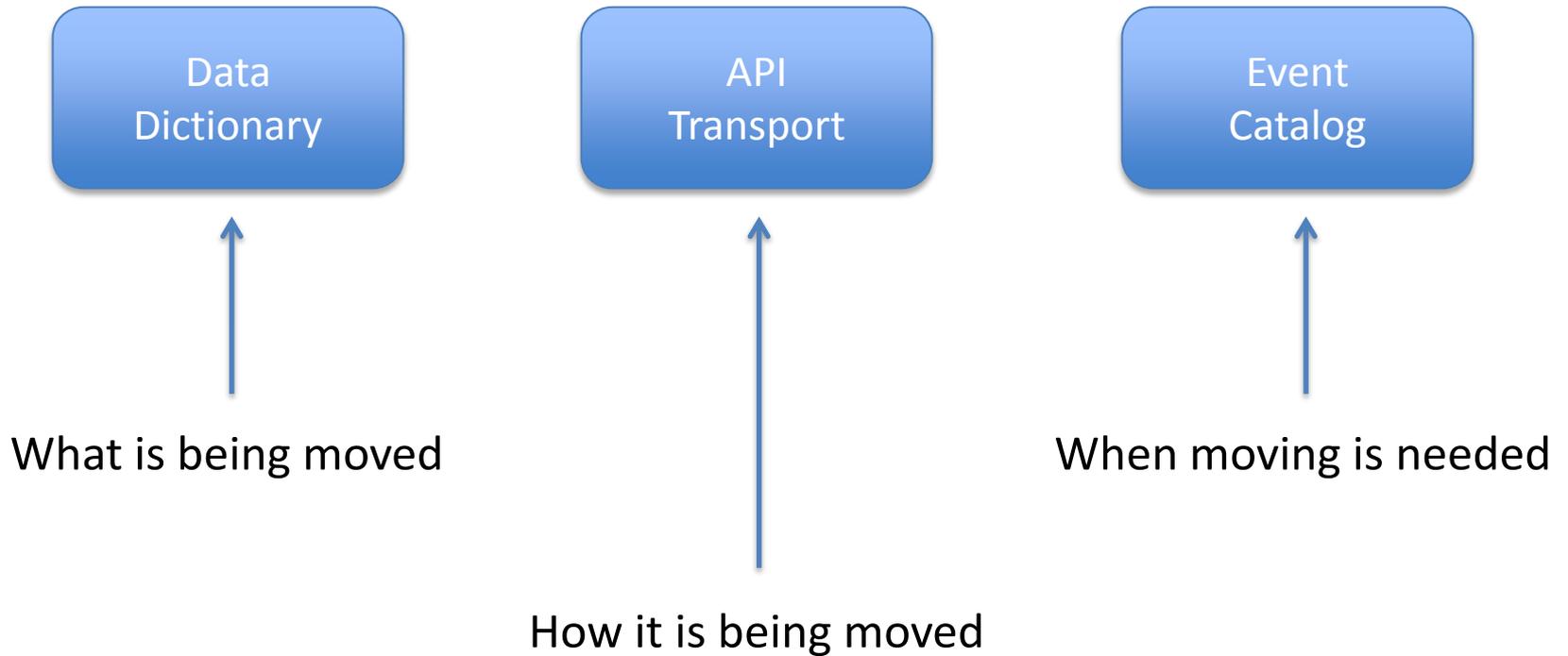
What are you talking about?

What is the Event Catalog?

PropertyDetail

- Actions performed by users

- Information to "fine tune" an action

- Data that results from an action

Events can Model a Life Cycle
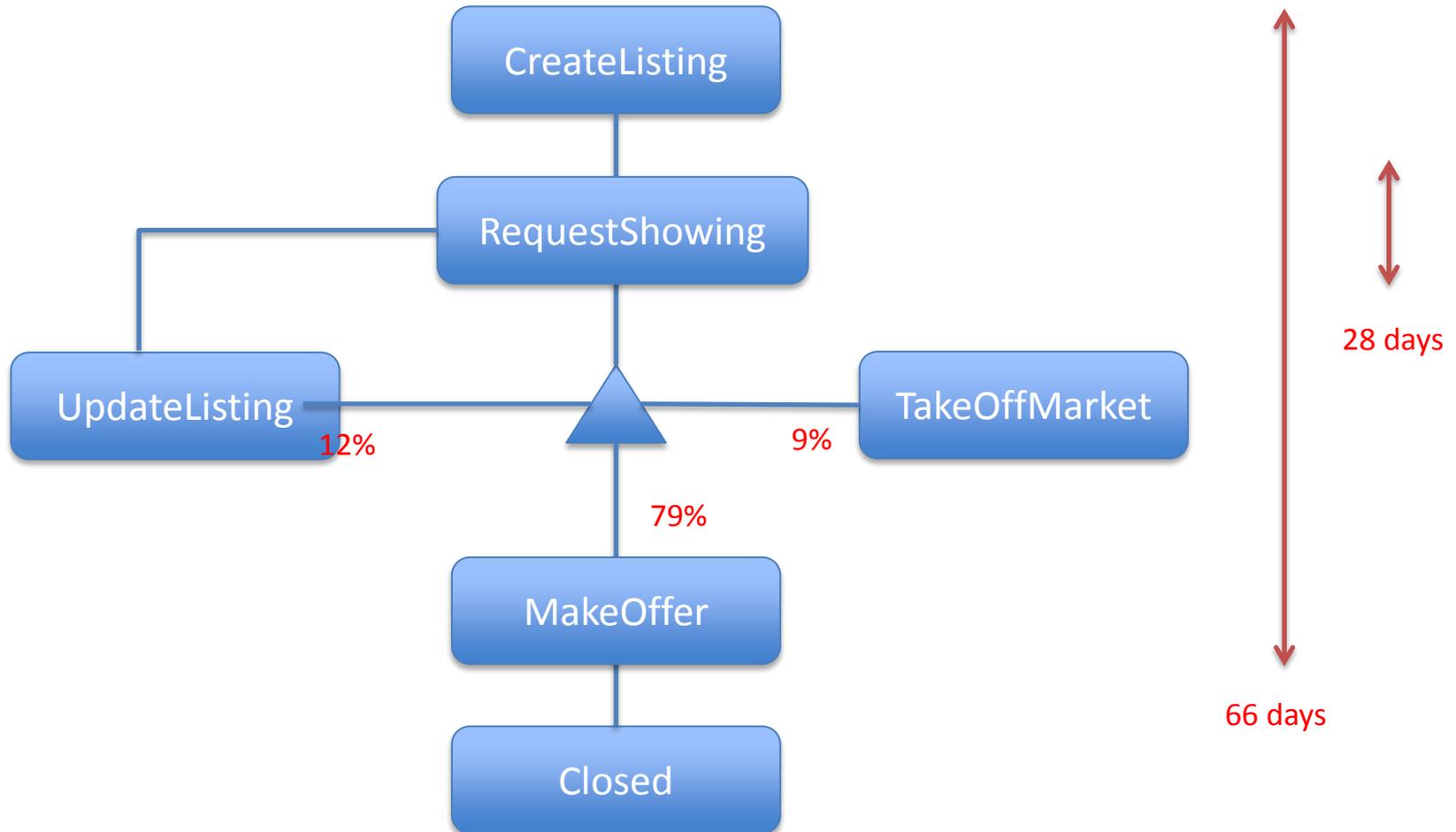
I need to be convinced!

# Why have an Event Catalog?

PropertyDetail

- Work can specified in a common way
- Easy to build simple applications
- Resource Consumption analysis

Resource Consumption
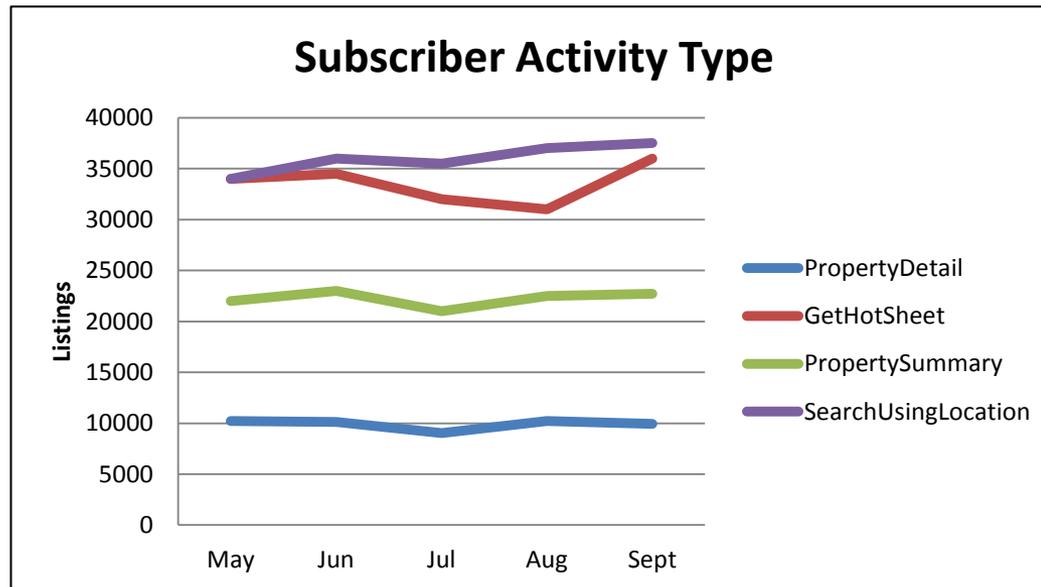
- Shows how users consume your services
- Helps new product/service development

Resource Activity
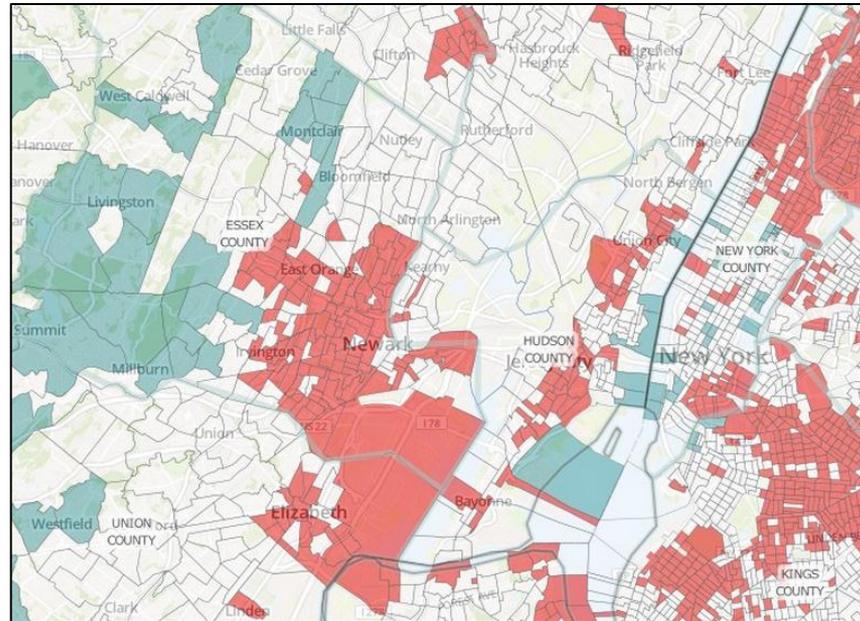
- Shows where a service is being used

- What is the next Event in the lifecycle?

- Predictive Analytics

Be more specific!

What an Event Catalog Looks Like

Data Dictionary Information
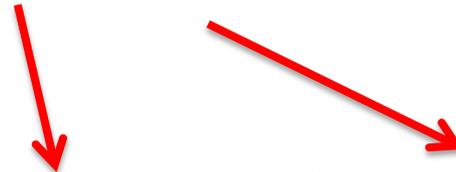
Event	Description	Variables	Response
PropertyDetail	Detail presentation of a listing suitable for public consumption	NONE	ListingNumber ListingPrice Area Bedrooms Baths Address
GetHotSheet	Summary of listings that have changed	SinceDate	ListingNumber NewPrice

What else could be in the Catalog?

- UpdateListing
- SearchUsingLocation
- PropertySummary
- RequestShowing
- MakeOffer

It should not be difficult to construct a list of activities

Give me an example!

Theoretical Construct

"Show me that"

Event: PropertyDetail
Address: 386 Sycamore

Request

"OK, here it is"

Event: PropertyDetail
Address: 386 Sycamore
Listing Price: $356,000
Bedrooms: 4
Baths: 3

Response

API

Data Dictionary

Event
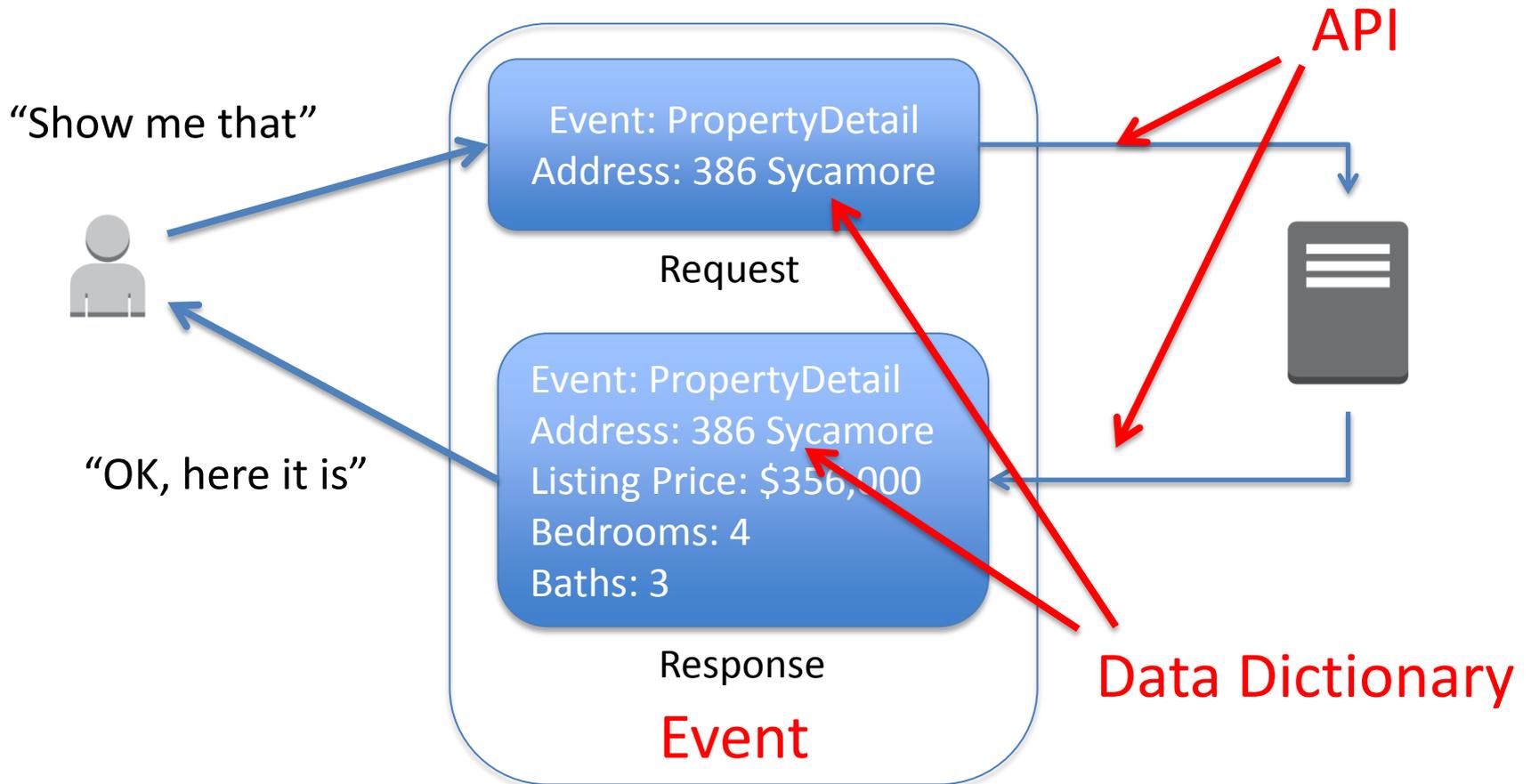
Theoretical Construct
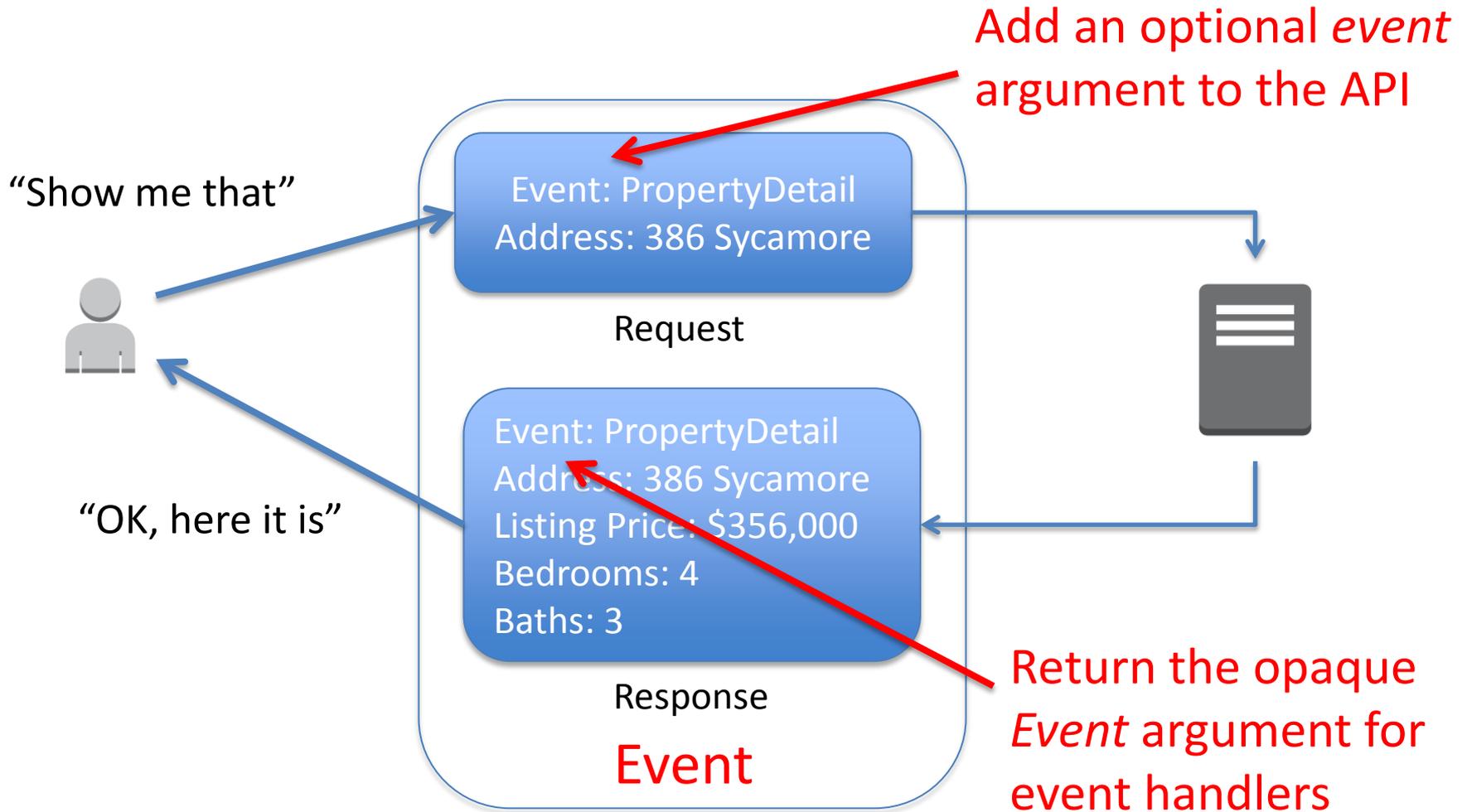
What Does Open Source Mean?

Do it Yourself!

DIY Processing Tools

Asynchronous programming languages for quicker data retrieval

Multi-threading javascript libraries for faster interactions

Images from respective communities listed

DIY Persistence Tools

Low overhead database that should be used where speed is an issue

Traditional SQL Database

NoSQL Database that works well with JSON structured data

# DIY Operating Systems

Low overhead, fast Unix

Enterprise grade
GNU/Linux

Debian GNU/Linux
(more packages)

Demo Schematic
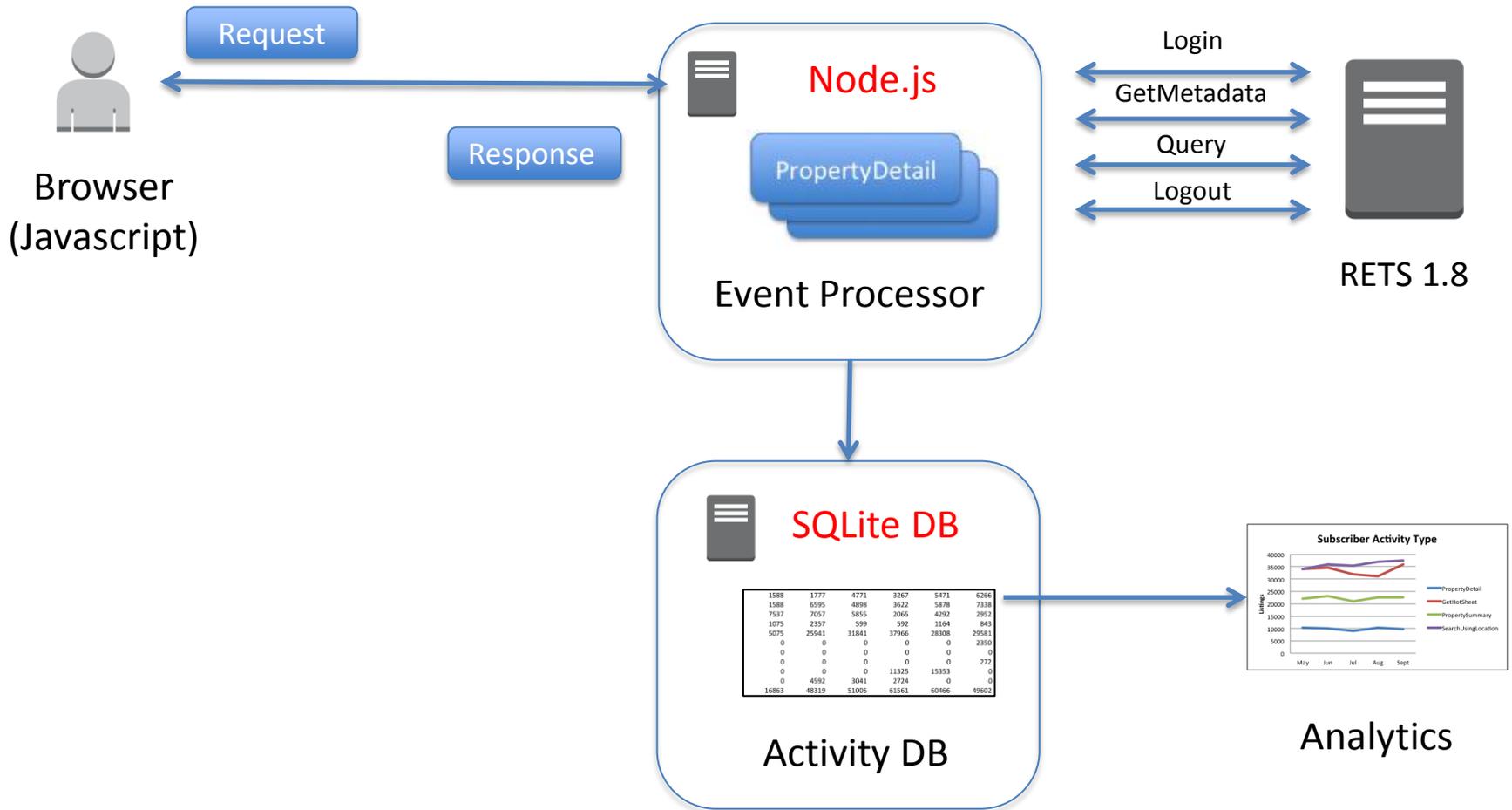
This didn't use the API!

RESO API w/Events: Fast

Browser
(Javascript)

Request

Response

RESO API

PropertyDetail

node js

mongoDB

freeBSD

Activity DB

1588	1777	4771	3267	5471	6266
1588	6595	4898	3622	5878	7338
7537	7057	5855	2065	4292	2952
1075	2357	599	592	1164	843
5075	25941	31841	37966	28308	29581
0	0	0	0	0	2350
0	0	0	0	0	0
0	0	0	0	0	272
0	0	0	11325	15353	0
0	4592	3041	2724	0	0
16863	48319	51005	61561	60466	49602

SQLite

Analytics

Images from respective communities listed

RESO API w/Events: High Volume

Request

Response

Browser
(Javascript)

RESO API Server

PropertyDetail

CentOS

ERLANG

MariaDB

Activity DB

CentOS

SQLite

1588	1777	4771	3267	5471	6266
1588	6595	4898	3622	5878	7338
7537	7057	5855	2065	4292	2952
1075	2357	599	592	1164	843
5075	25941	31841	37966	28308	29581
0	0	0	0	0	2350
0	0	0	0	0	0
0	0	0	0	0	272
0	0	0	11325	15353	0
0	4592	3041	2724	0	0
16863	48319	51005	61561	60466	49602

Analytics
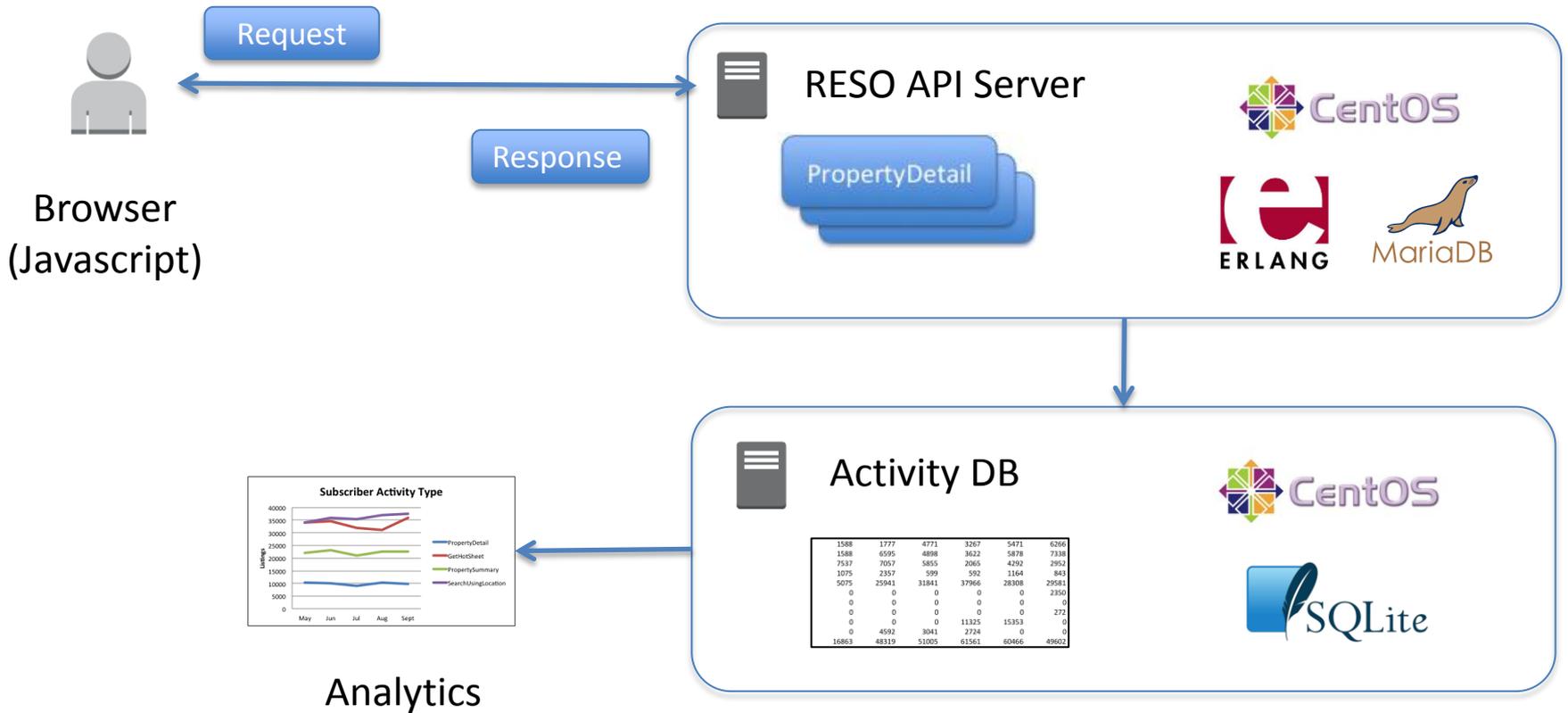
Images from respective communities listed

Relevance Achieved

Data
Dictionary

API
Transport

Event
Catalog