Using MongoDB Side-by-Side with RDBMS at Dealertrack

dealertrack technologies™

Dealertrack Technologies SaaS overview

■ We offer a SaaS and data products for the retail automotive supply chain

■ Customers include:

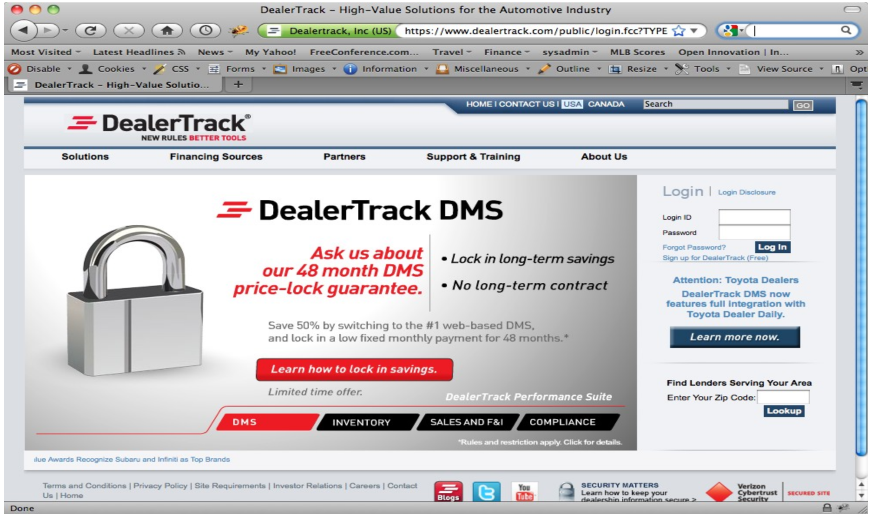
■ - 18,500 Dealers (New & Used)

■ - 6,000 Lenders

■ - 38 Manufacturers

■ - 500 Part Mfgrs & Insurers

■ - 14 State Governments



DealerTrack – High–Value Solutions for the Automotive Industry

Dealertrack, Inc (US) https://www.dealertrack.com/public/login.fcc?TYPE

DealerTrack DMS

Ask us about our 48 month DMS price-lock guarantee.

- Lock in long-term savings
- No long-term contract

Save 50% by switching to the #1 web-based DMS, and lock in a low fixed monthly payment for 48 months.*

Limited time offer.

*Rules and restriction apply. Click for details.

Attention: Toyota Dealers DealerTrack DMS now features full integration with Toyota Dealer Daily.

Find Lenders Serving Your Area

Enter Your Zip Code:

ılue Awards Recognize Subaru and Infiniti as Top Brands

SECURITY MATTERS Learn how to keep your dealership information secure >

■ Implemented with old & new tools:

■ - IIS / .NET / MSMQ / VB / C#

■ - Apache / Java / Python / Perl / Grails

■ - RPG2 / DDS / SQL

■ - Oracle / MS SQL Server / DB2 / MongoDB / MySQL / MUMPS

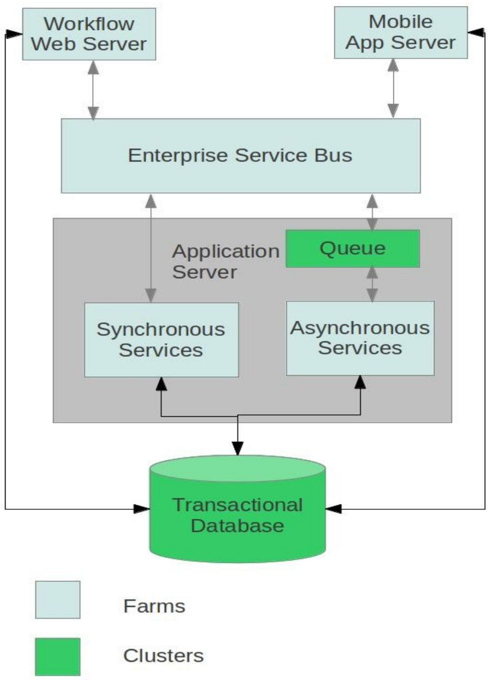
Core Solutions and Services Offered

- Transaction-based solutions for dealers and partners
  - Largest Online Credit Application Processing Network in the U.S. and Canada
  - Largest Outsourced Provider of Contract Processing in the U.S. and Canada
  - Contract and Vehicle Lien Processing for Lenders
  - Vehicle Registration and Titling Services for Dealers
  - Operate Private Web Solutions for Franchise Dealers (Honda, Mercedes, Nissan, Hyundai/Kia (Canada))
- Subscription-based solutions for dealers (Software-as-a-Service)
  - #1 Web-based Dealer Management System (DMS)
  - Industry Leading Sales And Finance Solutions
  - Desktop And Mobile Inventory Solutions
  - Largest Provider of Marketing and Interactive Solutions Including Web, Mobile, Search, Social And Chat
  - Unique Digital Retailing Solutions That No Other Industry Competitor Can Provide

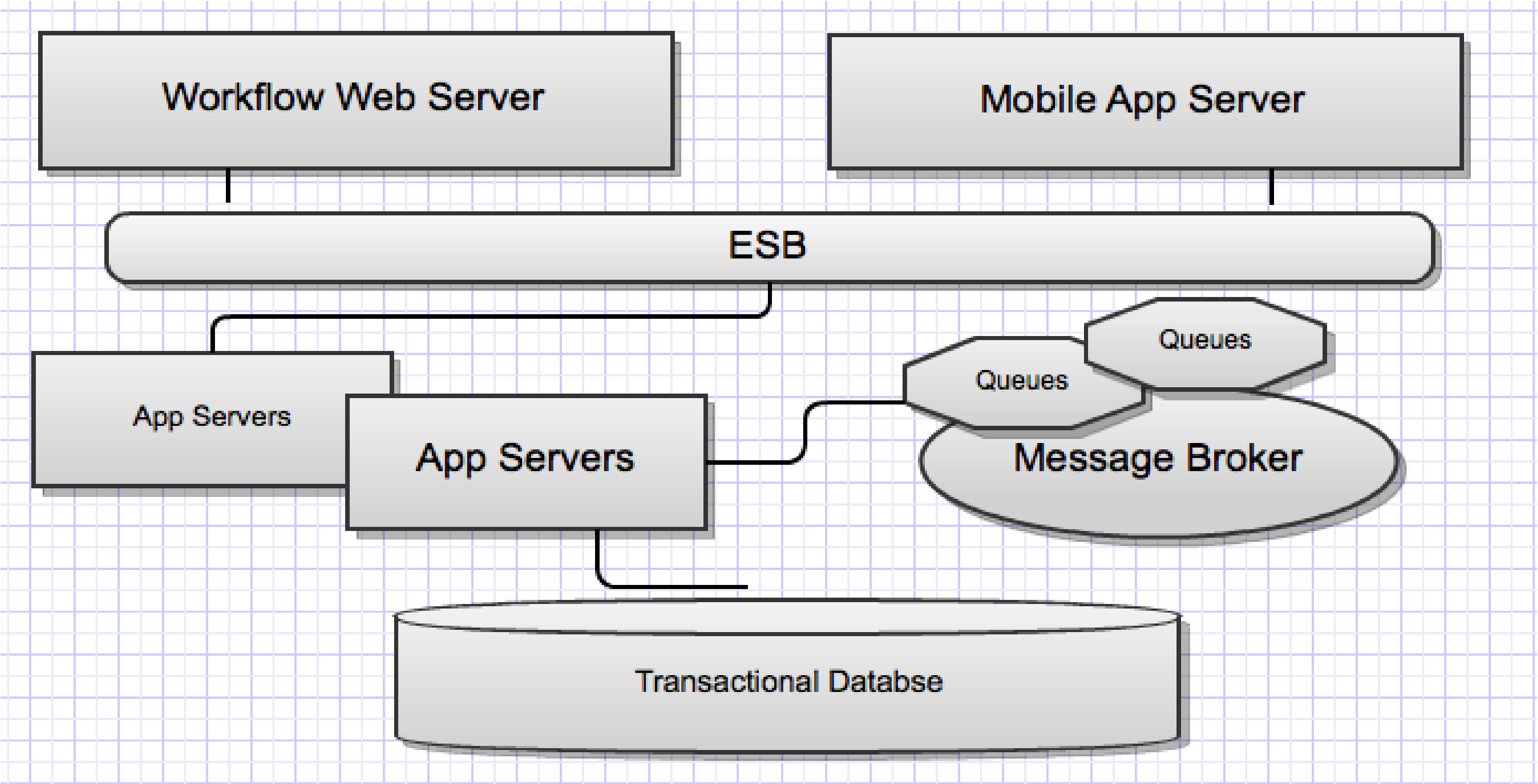
DealerTrack 2.0 Conceptual Architecture

Architecture Goals and Challenges

- Enable rapid growth in products and ser
- Doing away with point-to-point integratio
- “spaghetti code”
- Managing merger and acquisition driven
- growth
- Enabling applications or content for mult
- front-ends, including mobile devices
- Getting better operational visibility, mana integration / SOA infrastructure
- Legacy system modernization


Dealertrack Platform Topology

Dealertrack Platform Topology



Technology

- Platform: RedHat Linux
- Front-End: Python/Django/Apache WSGI
- Services layer – Python/Django/TastyPie
- Enterprise Service Bus – Mule ESB
- Messaging Broker – RedHat MRG
- Databases – Oracle, MongoDB, DB2, MySQL
- Local Cache - Redis

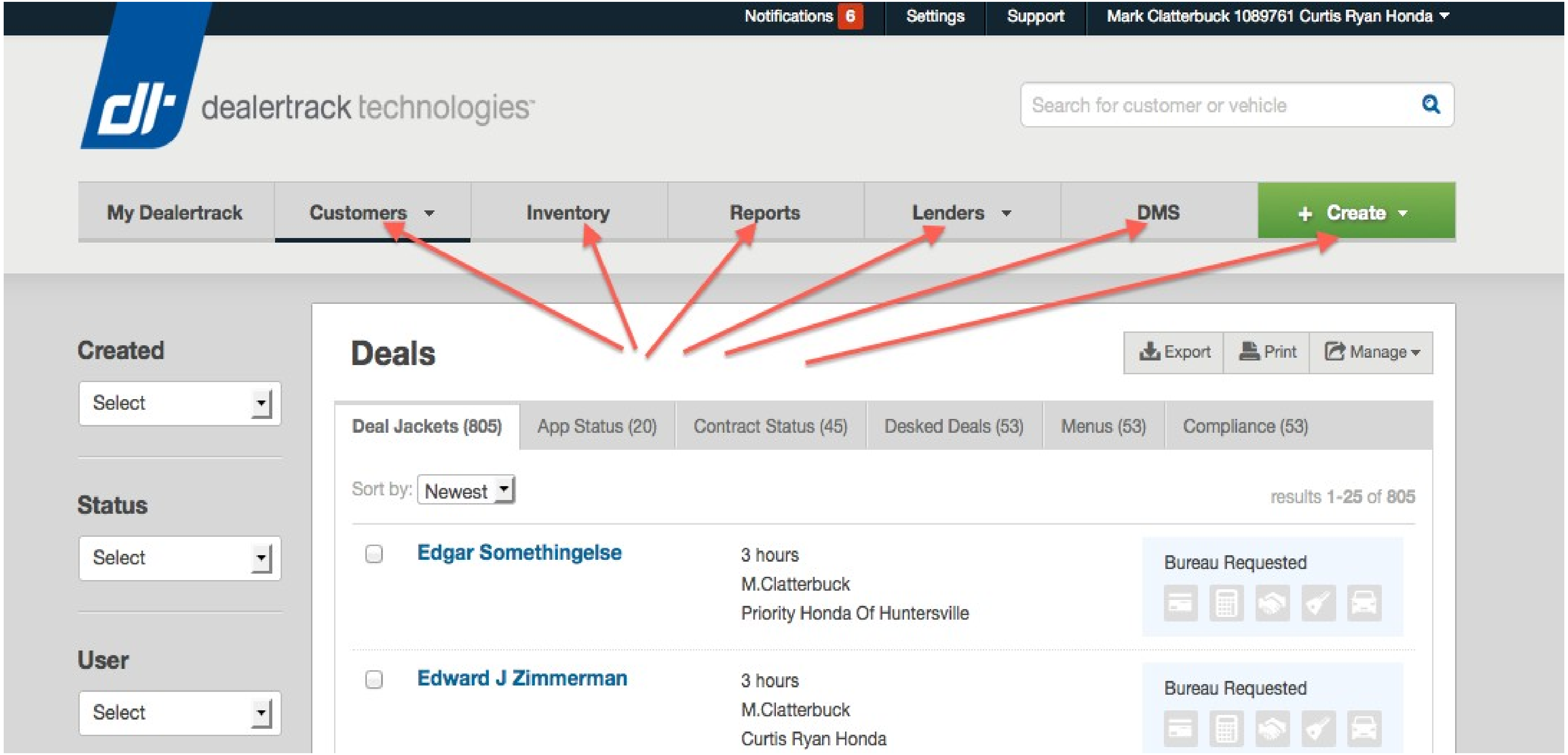
Case #1 – Using Mongo as HTTP Session backend

- Python
- Django
- Mongo DB

Why ?

- The application presents related but separate vertical functionality under one umbrella
- The shared features require their own schemas in the HTTP session but they all share same HTTP session
- Data needed in the HTTP session can vary by the specific functionality of the shared feature
- Performance is critical. We don't want to hold UI thread any more then we absolutely have to

This is what this web application UI looks like



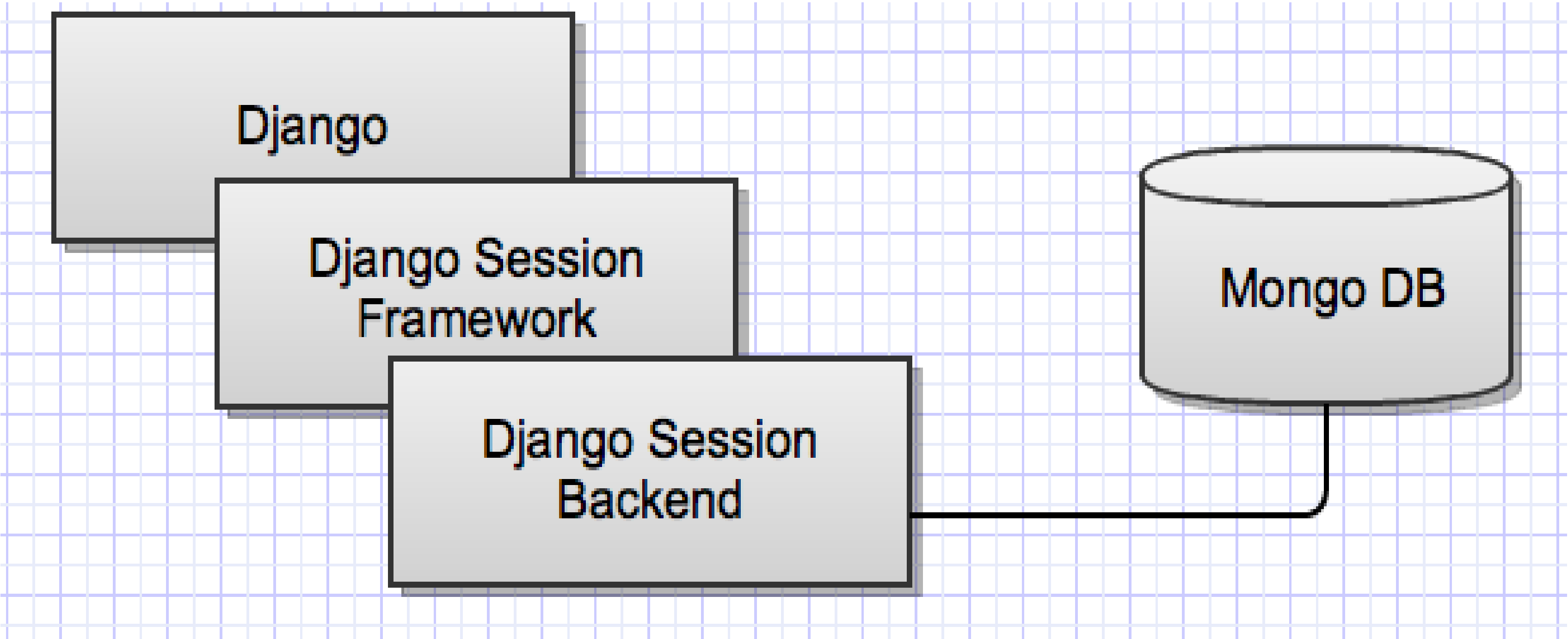
Deals

Sort by: Newest

results 1-25 of 805

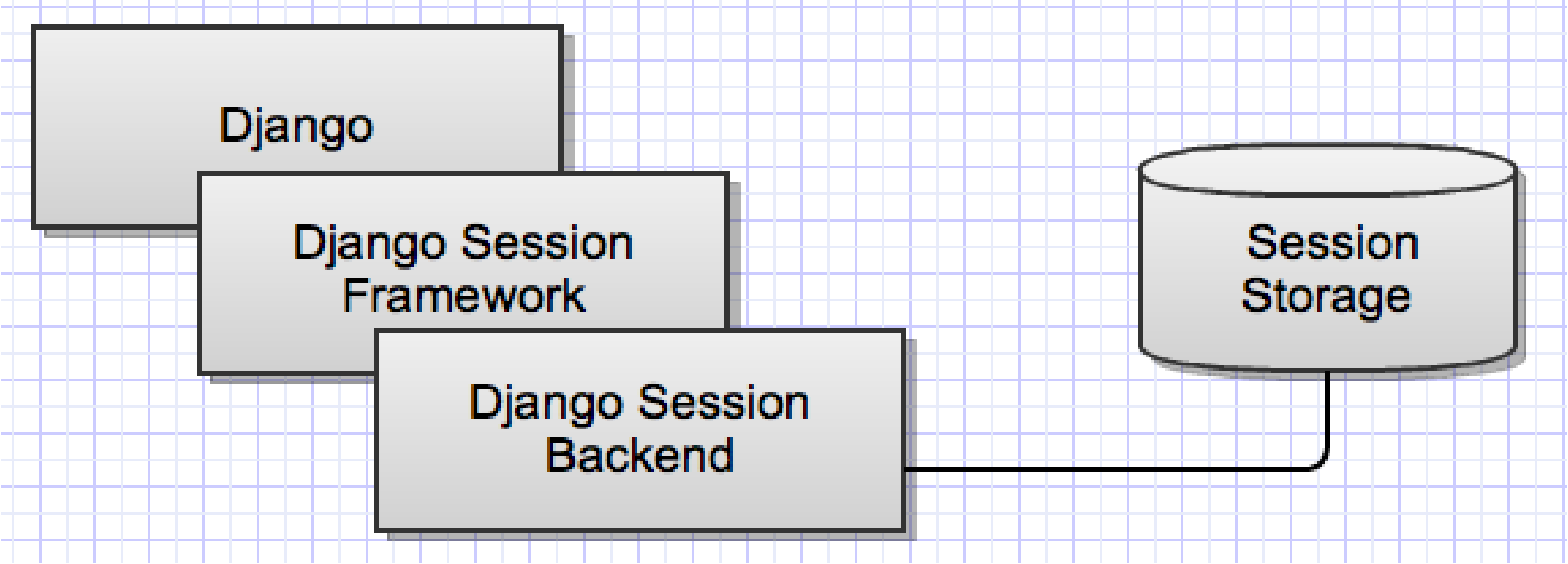
□	Edgar Somethingelse	3 hours M.Clatterbuck Priority Honda Of Huntersville	Bureau Requested
□	Edward J Zimmerman	3 hours M.Clatterbuck Curtis Ryan Honda	Bureau Requested

Introducing MongoDB as session storage



It is a question of responsibilities

- Need to know basis
- Front-End does NOT need to know where the session data is stored



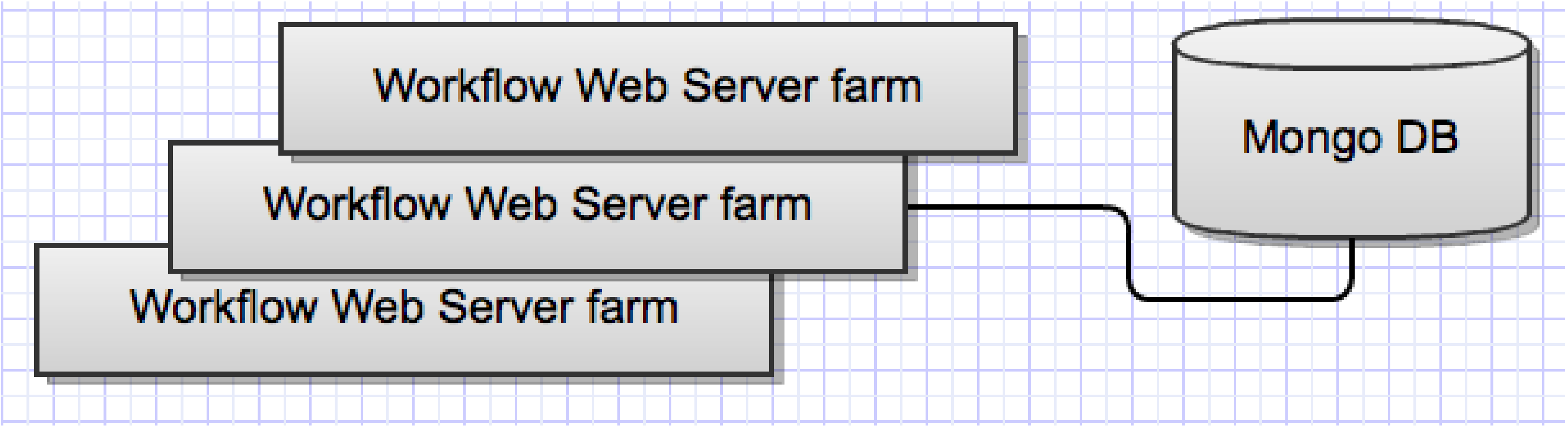
What do we like about this?

- Allows for the the shared features to use their own schemas in the HTTP session while sharing the HTTP session
- Allows the data to vary by the specific functionality of the shared feature
- Front End is completely oblivious of where the data actually stored
- Using MongoDB – obviously cool

Is this the optimal solution?

- The session data is stored in a central datastore
- There is some local area network overhead writing into remote centralized MongoDB

- What do you think? Any ideas, suggestions? Lets share some thoughts

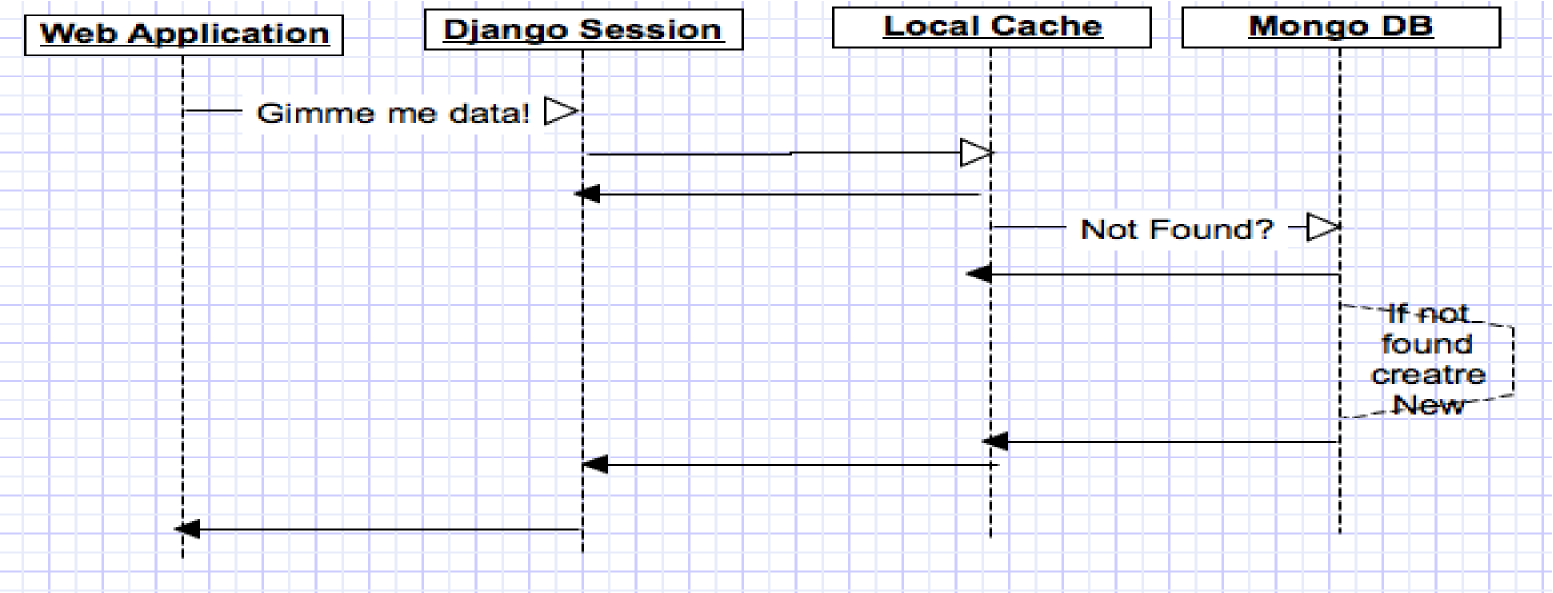


It beats our old solution, but can we make it even better?

- We like a lot of this approach
- We have one concern – remote centralized database calls for every session request
- Why don't we optimize?
  - MongoDB is 4x faster than Oracle w/ 10% the capacity cost
  - It is still slower than reading and writing from local memory
  - It is still important for the front end not to know anything about it
- What can we do?
- Introduce local cache – we are using redis

Enter local cache

- If the data is not present in the local cache – read it from Mongo and then cache it locally
- If the data is not in Mongo – create new session and return



Redis local cache + MongoDB centralized session store

- Sessions are sticky to the web servers
- Session and reference data stored in local cache and mostly accessed from there
- Data also stored in Mongo as a master cache backend

- redis = Local Session Cache
- MongoDB – Centralized Session Persistence

- Custom Django session backend makes it all transparent to the front end web application
- Custom redis session backend connects local redis with centralized Mongo

Case #2 – Using Mongo DB to cache relational data in Inventory+

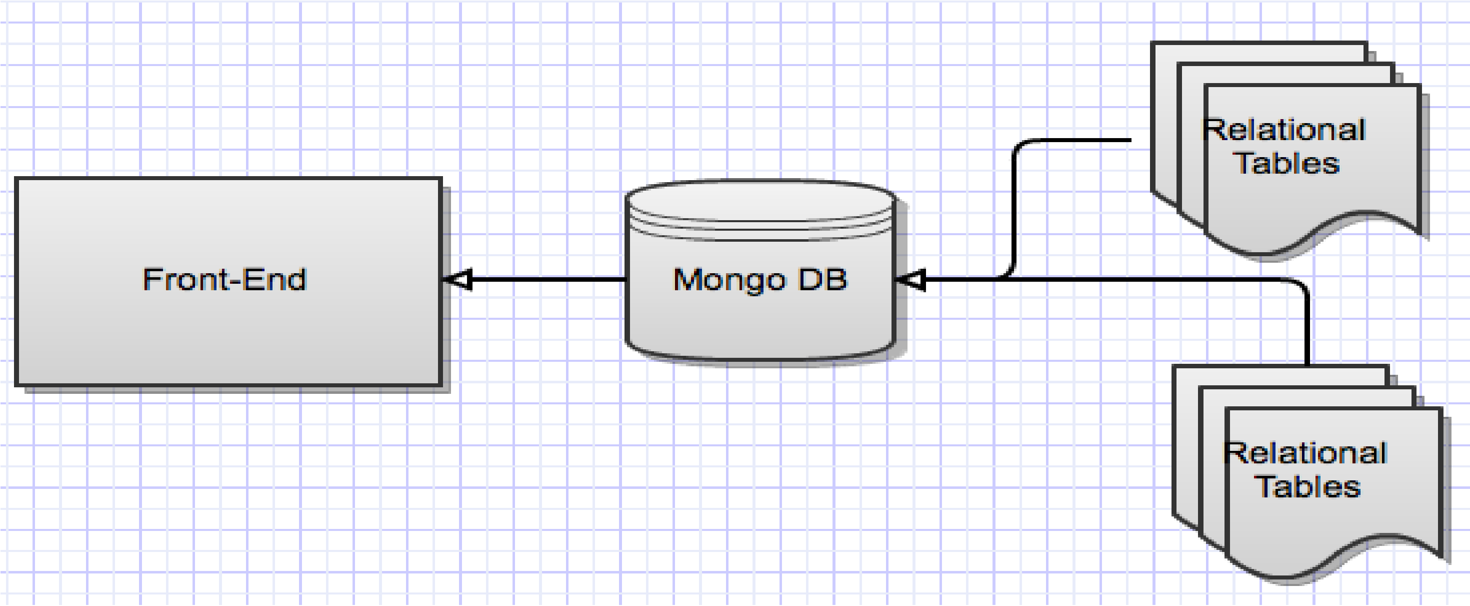
- Inventory+ – another Dealertrack Technologies Solution for auto dealers

Major Features

- Vehicle Inventory Management – Marketing, Pricing, Aging, etc…
- Pricing Analytics – Price your inventory based on real market data
- Dealership Websites – Based on vehicle inventory and updated in real time
- Chat – Host and manage chat services for dealership websites
- Many more features for our dealer users

Case #2 – Using Mongo DB to cache vehicle data in Inventory+

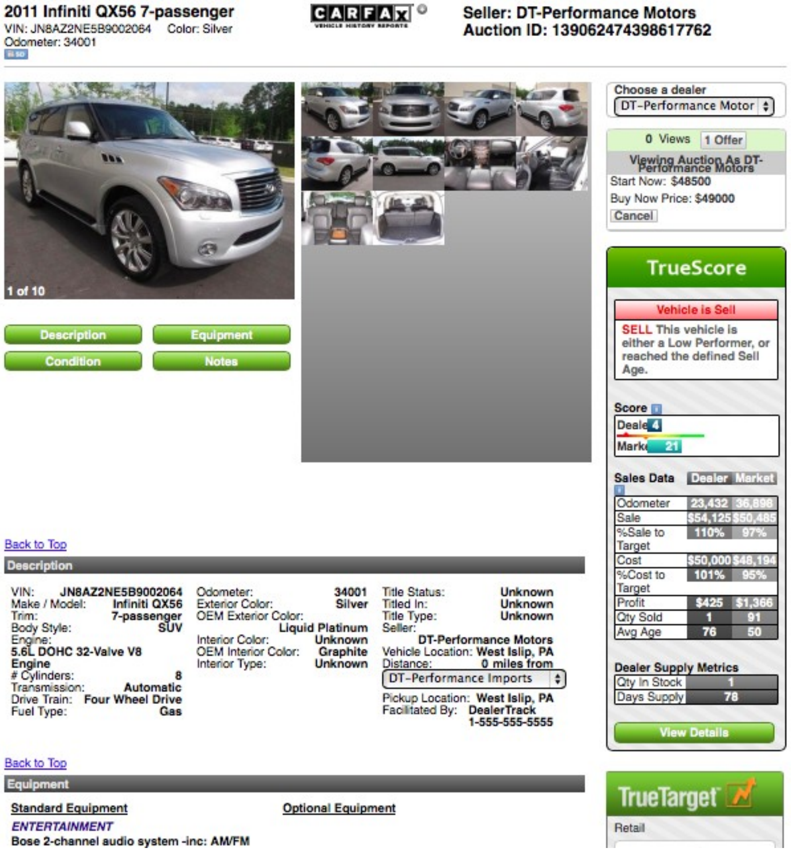
- eCarList is using Mongo DB in to cache large amounts of relational data and make it searchable



UI View

Vehicle Attributes cached and read from MongoDB

- VIN
- Year, Make, Model, etc…
- Vehicle Photo URLs
- Attributes
    - Odometer
    - Color
    - Transmission
    - Engine
    - Etc…
- Optional Equipment
- Standard Equipment
- Pricing Info & Price History



2011 Infiniti QX56 7-passenger
VIN: JN8AZ2NE5B9002064   Color: Silver
Odometer: 34001

Seller: DT-Performance Motors
Auction ID: 1390624743986177762

Choose a dealer
DT–Performance Motor

0 Views 1 Offer
Viewing Auction As DT-Performance Motors
Start Now: **$48500**
Buy Now Price: **$49000**

Vehicle is Sell

SELL This vehicle is either a Low Performer, or reached the defined Sell Age.

Score
Deale 4
Marke 21

Sales Data	Dealer	Market
Odometer	23,432	36,898
Sale	$54,125	$50,485
%Sale to Target	110%	97%
Cost	$50,000	$48,194
%Cost to Target	101%	95%
Profit	$425	$1,366
Qty Sold	1	91
Avg Age	76	50

Dealer Supply Metrics

Qty In Stock	1
Days Supply	78

Back to Top

Description

VIN:	**JN8AZ2NE5B9002064**	Odometer:	**34001**	Title Status:	**Unknown**
Make / Model:	**Infiniti QX56**	Exterior Color:	**Silver**	Titled In:	**Unknown**
Trim:	**7-passenger**	OEM Exterior Color:	**Liquid Platinum**	Title Type:	**Unknown**
Body Style:	**SUV**	Interior Color:	**Unknown**	Seller:	**DT-Performance Motors**
Engine:	**5.6L DOHC 32-Valve V8 Engine**	OEM Interior Color:	**Graphite**	Vehicle Location:	**West Islip, PA**
# Cylinders:	**8**	Interior Type:	**Unknown**	Distance:	**0 miles from** DT–Performance Imports
Transmission:	**Automatic**			Pickup Location:	**West Islip, PA**
Drive Train:	**Four Wheel Drive**			Facilitated By:	**DealerTrack** **1-555-555-5555**
Fuel Type:	**Gas**				

Back to Top

Equipment

Standard Equipment   **Optional Equipment**

ENTERTAINMENT
Bose 2-channel audio system -inc: AM/FM

General Use Case

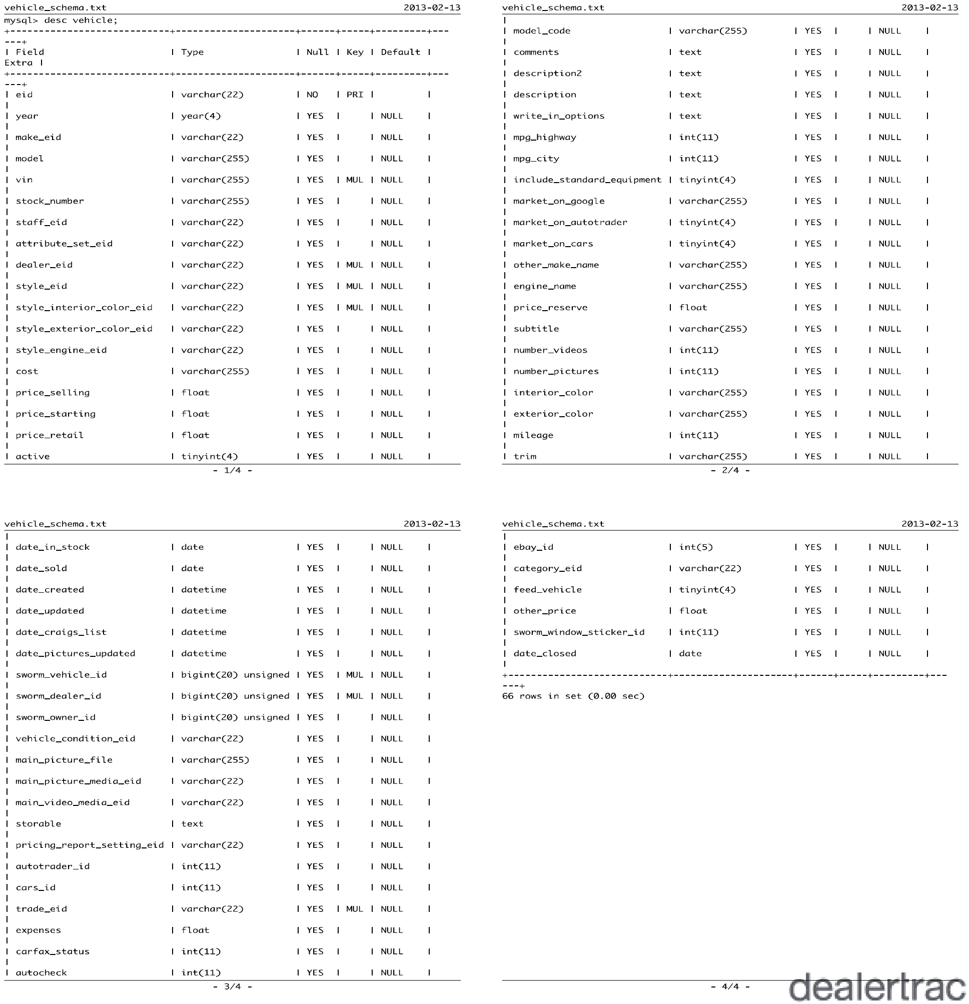
- The objects that eCarList manages are built around Perl’s strengths: arrays and hashes
- Relational databases do not afford a one-for-one mapping of these data structures, so storing and retrieving them is a challenge
- Mongo natively handles these structures, and therefore presents a cleaner interface for storing and retrieving Perl objects
- If a Perl query is conforming to Mongo’s indexing, lookups are nearly O(1)
- MongoDB is preloaded with search results, further extending the O(1) lookup time

Advantages of Mongo over MySQL or DB2 in Inventory+

Vehicle MySQL Schema

This table is linked to twenty-two additional tables.

Caching assembled objects in MongoDB pays off



```
vehicle_schema.txt                                              2013-02-13
mysql> desc vehicle;
+----------------------------+---------------------+------+-----+---------+---
---+
| Field                      | Type                | Null | Key | Default |
Extra |
+----------------------------+---------------------+------+-----+---------+---
---+
| eid                        | varchar(22)         | NO   | PRI |         |
|
| year                       | year(4)             | YES  |     | NULL    |
|
| make_eid                   | varchar(22)         | YES  |     | NULL    |
|
| model                      | varchar(255)        | YES  |     | NULL    |
|
| vin                        | varchar(255)        | YES  | MUL | NULL    |
|
| stock_number               | varchar(255)        | YES  |     | NULL    |
|
| staff_eid                  | varchar(22)         | YES  |     | NULL    |
|
| attribute_set_eid          | varchar(22)         | YES  |     | NULL    |
|
| dealer_eid                 | varchar(22)         | YES  | MUL | NULL    |
|
| style_eid                  | varchar(22)         | YES  | MUL | NULL    |
|
| style_interior_color_eid   | varchar(22)         | YES  | MUL | NULL    |
|
| style_exterior_color_eid   | varchar(22)         | YES  |     | NULL    |
|
| style_engine_eid           | varchar(22)         | YES  |     | NULL    |
|
| cost                       | varchar(255)        | YES  |     | NULL    |
|
| price_selling              | float               | YES  |     | NULL    |
|
| price_starting             | float               | YES  |     | NULL    |
|
| price_retail               | float               | YES  |     | NULL    |
|
| active                     | tinyint(4)          | YES  |     | NULL    |
                                 - 1/4 -
```

```
vehicle_schema.txt                                              2013-02-13
|
| model_code                 | varchar(255)        | YES  |     | NULL    |
|
| comments                   | text                | YES  |     | NULL    |
|
| description2               | text                | YES  |     | NULL    |
|
| description                | text                | YES  |     | NULL    |
|
| write_in_options           | text                | YES  |     | NULL    |
|
| mpg_highway                | int(11)             | YES  |     | NULL    |
|
| mpg_city                   | int(11)             | YES  |     | NULL    |
|
| include_standard_equipment | tinyint(4)          | YES  |     | NULL    |
|
| market_on_google           | varchar(255)        | YES  |     | NULL    |
|
| market_on_autotrader       | tinyint(4)          | YES  |     | NULL    |
|
| market_on_cars             | tinyint(4)          | YES  |     | NULL    |
|
| other_make_name            | varchar(255)        | YES  |     | NULL    |
|
| engine_name                | varchar(255)        | YES  |     | NULL    |
|
| price_reserve              | float               | YES  |     | NULL    |
|
| subtitle                   | varchar(255)        | YES  |     | NULL    |
|
| number_videos              | int(11)             | YES  |     | NULL    |
|
| number_pictures            | int(11)             | YES  |     | NULL    |
|
| interior_color             | varchar(255)        | YES  |     | NULL    |
|
| exterior_color             | varchar(255)        | YES  |     | NULL    |
|
| mileage                    | int(11)             | YES  |     | NULL    |
|
| trim                       | varchar(255)        | YES  |     | NULL    |
                                 - 2/4 -
```

```
vehicle_schema.txt                                              2013-02-13
|
| date_in_stock              | date                | YES  |     | NULL    |
|
| date_sold                  | date                | YES  |     | NULL    |
|
| date_created               | datetime            | YES  |     | NULL    |
|
| date_updated               | datetime            | YES  |     | NULL    |
|
| date_craigs_list           | datetime            | YES  |     | NULL    |
|
| date_pictures_updated      | datetime            | YES  |     | NULL    |
|
| sworm_vehicle_id           | bigint(20) unsigned | YES  | MUL | NULL    |
|
| sworm_dealer_id            | bigint(20) unsigned | YES  | MUL | NULL    |
|
| sworm_owner_id             | bigint(20) unsigned | YES  |     | NULL    |
|
| vehicle_condition_eid      | varchar(22)         | YES  |     | NULL    |
|
| main_picture_file          | varchar(255)        | YES  |     | NULL    |
|
| main_picture_media_eid     | varchar(22)         | YES  |     | NULL    |
|
| main_video_media_eid       | varchar(22)         | YES  |     | NULL    |
|
| storable                   | text                | YES  |     | NULL    |
|
| pricing_report_setting_eid | varchar(22)         | YES  |     | NULL    |
|
| autotrader_id              | int(11)             | YES  |     | NULL    |
|
| cars_id                    | int(11)             | YES  |     | NULL    |
|
| trade_eid                  | varchar(22)         | YES  | MUL | NULL    |
|
| expenses                   | float               | YES  |     | NULL    |
|
| carfax_status              | int(11)             | YES  |     | NULL    |
|
| autocheck                  | int(11)             | YES  |     | NULL    |
                                 - 3/4 -
```

```
vehicle_schema.txt                                              2013-02-13
|
| ebay_id                    | int(5)              | YES  |     | NULL    |
|
| category_eid               | varchar(22)         | YES  |     | NULL    |
|
| feed_vehicle               | tinyint(4)          | YES  |     | NULL    |
|
| other_price                | float               | YES  |     | NULL    |
|
| sworm_window_sticker_id    | int(11)             | YES  |     | NULL    |
|
| date_closed                | date                | YES  |     | NULL    |
|
+----------------------------+---------------------+------+-----+---------+---
---+
66 rows in set (0.00 sec)

                                 - 4/4 -
```

Mongo Overview in Inventory+

- MongoDB used to cache Vehicle related data from associated tables
- Single lookup by EID, or DEALER_EID allows for simple ways to get at Vehicle documents
- Previous methods to access data required multiple table lookups, or complex JOIN SQL

```
mongos> db.vehicle.find({'vehicle_eid':'59VpEPG8F2I6vUO4Uh1HCw'}).pretty()
{
        "vin" : "SCFFDCCD2BGE12420",
        "year" : 2011,
        "make_eid" : "Cc0Jlpb5gi5lc1RFgbh+iw",
        "make" : "Aston Martin”,
        "model" : "DBS Volante",
        "selling_price" : 192888,
        "engine_name" : "5.9L DOHC 48-Valve V12 Engine",
        "vehicle_attributes" : {
                "transmission" : "Automatic",
                "drivetrain" : "Rear Wheel Drive",
                "body_type" : "Convertible"
                …
        "main_photo" : {
                "photos" : {
                        "640" : "http://photos.ecarlist.com/jS/Qn/Rl/4k/8y/Nw/9w/WE/xP/2c/pA_640.jpg",
                        "original" : "http://photos.ecarlist.com/jS/Qn/Rl/4k/8y/Nw/9w/WE/xP/2c/pA.jpg",
                        ...
```

Comparison of Features

Relational Databases

- Hundreds of columns per row
- Referential integrity preserves data quality from central definitions
- SQL Native Format conversion to Perl associative array
- No easy way to cluster servers
- Inner join operations are I/O blocking, slowing down retrievals and forcing sequential access of large objects
- Update efficiency for normalized data models

MongoDB

- Nested JSON arrays of arrays
- JSON values returned by Mongo in ‘native’ Perl format
- Server clustering built in
- No inner joins required, retrievals are ‘atomic’ and very fast

General Use Case

- Given a unique identifier, retrieve all information associated with a Vehicle in nested form
- MySQL cannot do this in a single step, even with a large number of embedded inner join operations.
- We have now moved the relational tables to a high-performance DB2 cluster
- Whenever any of the attributes of a vehicle are updated (either in batch or by an interactive user) MongoDB is updated as the DB2 transaction commits
- MongoDB returns (via JSON) a complete Perl object, including internal classes with instance values for any requested Vehicle ID

Batch Data Feed Management

- Our customers and partners send us thousands of files (large and small) multiple times per day with a variety vehicle inventory information for analytical processing and for preparation for web listing (on dealer websites and portals)
- We have a complex workflow to process the contents of these files and update the information in our databases, recalculate analytics, reformat information, and update our managed websites as well as send the data out to other systems
- Decomposition of incoming jobs into file-by-file parallel executions requires complex scheduling and tracking of the state of processing to control scheduling of parallel execution
- MongoDB provides the speed of a queuing system and the multi-index inquiry flexibility of a DBMS (this is a one-table/one-collection data model) for managing the flows.

Three MongoDB use/cases at Dealertrack:

Distributed back-end for an optimistic local cache

High-speed key:value datastore for complex objects

Batch data feed management

Next Up: Backing store for AMQP/QPID/MRG

Discussion and Questions

We are growing and hiring!

www.dealertrack.com/portal/careers-home