Legacy monolith to microservices
● Software Engineer at Dubizzle
● Founder of Zypher
● Open source organizations I have been part of
  ○ Gluster
  ○ Ngui
● Former Chapter Lead of GDG Gandhinagar
● Github: github.com/smitthakkar96
● Linkedin: linkedin.com/in/smitthakkar96
● https://zypher.tech
Dinosaur skeleton on sale for $2.4m by Dubai website

Brontosaurus dinosaur skeleton is 85% complete and was discovered 10 years ago in the US.

Dinosaur fans were in for a surprise recently when a skeleton of a Brontosaurus was put up for sale for AED82 million ($22.9 million) on Dubai classified listings site Dubizzle.
Empower teams to sustainably deliver value
Choose your Dubizzle City:

- Abu Dhabi
- Dubai

more cities coming soon!
if site_profile.country_site.id == 4:  # UAE
    facebook_url = 'http://www.facebook.com/dubizzle'
elif site_profile.country_site.id == 7:  # Egypt
    facebook_url = 'http://www.facebook.com/DubizzleEgypt'
elif site_profile.country_site.id == 10:  # KSA
    facebook_url = 'http://www.facebook.com/DubizzleKSA'
elif site_profile.country_site.id == 17:  # Kuwait
    facebook_url = 'http://www.facebook.com/pages/Dubizzle-Kuwait/133537483367816'
elif site_profile.country_site.id == 19:  # Qatar
    facebook_url = 'http://www.facebook.com/pages/Dubizzle-Qatar/144285568954101'
elif site_profile.country_site.id == 21:  # Bahrain
    facebook_url = 'http://www.facebook.com/pages/Dubizzle-Bahrain/176168439062742'
elif site_profile.country_site.id == 23:  # Oman
elif site_profile.country_site.id == 25:  # Syria
    facebook_url = 'http://www.facebook.com/DubizzleSyria'
elif site_profile.country_site.id == 27:  # Jordan
    facebook_url = 'http://www.facebook.com/DubizzleJordan'
elif site_profile.country_site.id == 29:  # Lebanon
    facebook_url = 'http://www.facebook.com/DubizzleLebanon'
elif site_profile.country_site.id == 31:  # Algeria
    facebook_url = 'http://www.facebook.com/dubizzle'
elif site_profile.country_site.id == 33:  # Libya
    facebook_url = 'http://www.facebook.com/dubizzle'
elif site_profile.country_site.id == 37:  # Morocco
    facebook_url = 'http://www.facebook.com/pages/Dubizzle-Morocco/112728338792888'
<table>
<thead>
<tr>
<th>OPERATION</th>
<th>SQL</th>
<th>HTTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>INSERT</td>
<td>PUT/POST</td>
</tr>
<tr>
<td>Read</td>
<td>SELECT</td>
<td>GET</td>
</tr>
<tr>
<td>Update</td>
<td>UPDATE</td>
<td>PUT/PATCH</td>
</tr>
<tr>
<td>Delete</td>
<td>DELETE</td>
<td>DELETE</td>
</tr>
</tbody>
</table>
POST /ad
{ "title": "Lorem Ipsum",
  "description": "...

PUT /ad/xyz/feature/
{ "user_id": 123, "status": true

POST /subscription
{ "user_id": 123, "status": "inactive"}
Presentation Layer

Logic Layer

Data Layer

Feature X
If you remake awful software from scratch without changing the culture that created it: you'll remake awful software
The Guiding Principles

★ Low Coupling, High Cohesion
★ No service can depend on another for uptime, functionality or data availability
★ One transaction can span at most one service
★ No vertical can impact the stability of another
★ Bounded contexts are defined by business realms not CRUD APIs
★ Choreography over orchestration
★ Events are a first class citizen
★ Analytics and Logs are the only source of debugging information
★ Rate limits, data privacy, authentication and authorization are mandatory for all services
★ Every service must be auto-deployable independently, after passing unit, functional, load and security tests
Low Coupling, High Cohesion

Ad creation context:
- Transactions
- Payments
- Moderation
- Products
- Ads
- Users

Ad consumption context:
- Report Ad
- Favourites
- Ads
- Users

Ad creation context:
- Products
- Ads
- Users

Ad consumption context:
- Report Ad
- Favourites
- Ads
- Users
No service can depend on another for uptime, functionality or data availability.
★ One transaction can span at most one service
“Any organization that designs a system (defined broadly) will … produce a design whose structure is a copy of the organization’s communication structure.”

- Melvin Conway

“If you have four groups working on a compiler, you’ll get a 4-pass compiler”

- Eric S Raymond
★ Bounded contexts are defined by business realms not CRUD APIs
Message Bus
Observability
We use newrelic for application performance monitoring, Error reporting, analytics and alerts.

Newrelic offers us an ability to do distributed tracing where synchronous transactions are happening.

New relic alerts are routed to pagerduty which raises escalation to the owners of the respective components.
All the application and system logs are propagated to Elasticsearch via syslog and logstash. Some alerts are also configured on top of the logs we get and are sent to the owners of the component via PagerDuty and Slack.
Infrastructure monitoring

- SRE Team (on call)
- Devs (on call)
- pagerduty
Crons and Background tasks
Queue Alarms

CloudAMQP

Total queue length alarm

Queue length alarm triggered

https://hilarious-crow.rmq.cloudamqp.com/#/

<table>
<thead>
<tr>
<th>Queue</th>
<th>Vhost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>Threshold</td>
</tr>
<tr>
<td>113166</td>
<td>50000</td>
</tr>
<tr>
<td>Time threshold</td>
<td>Instance name</td>
</tr>
<tr>
<td>60 minutes</td>
<td>Production</td>
</tr>
<tr>
<td>Cluster name</td>
<td>hilarious-crow</td>
</tr>
</tbody>
</table>
Cron Jobs

infra-alerts  APP  11:13 PM

[Alerting] uae-greedy-unpromote-motors-ads alert
@channel
chronos.chronos_job/uae-greedy-unpromote-motors-ads/status
0.95555555555556

Grafana v4.6.3  |  Yesterday at 11:13 PM
Meet telemetry
Measure timelag for messages with telemetry
Measure execution time for messages with telemetry
Monitor the size of messages with telemetry
Monitor throughput with telemetry
Monitor crons using telemetry
Telemetry also provides us low level APIs to record application specific metrics so that we can create dashboards with those metrics using telemetry.
from telemetry.api import get_client

SERVICE_NAME = 'my-project'
TELEMETRY_URL = 'influxdb://root:root@localhost/db

client = get_client(SERVICE_NAME, TELEMETRY_URL)

def your_func(*args, **kwargs):
    ...
    client.record_point(
        metric='send_emails.seller_emails',  # required
        value=seller_msg_count,  # default: value=1
    )
from telemetry.api import get_client

SERVICE_NAME = 'my-project'
TELEMETRY_URL = 'influxdb://root:root@localhost/db

client = get_client(SERVICE_NAME, TELEMETRY_URL)

def your_func(*args, **kwargs):
    ...
    client.record_data(
        metric='send_emails.seller_emails', # required
data={
            'value': seller_msg_count,
            'event_id': uuid.uuid4().hex,
        }, # required
    )
Bounded Context

Message Bus

Eventual Consistency

Support Tooling

Event-based architecture

Developer Training

 dubbedle
Questions?
Thank You!
We are hiring!

- Python Engineers
- Full Stack Engineers
- Mobile Engineers

Let’s get in touch on smit.thakkar@dubizzle.com