If your API is the answer, could you please rephrase the question?

Ernst Naezer & Flavia Sequeira

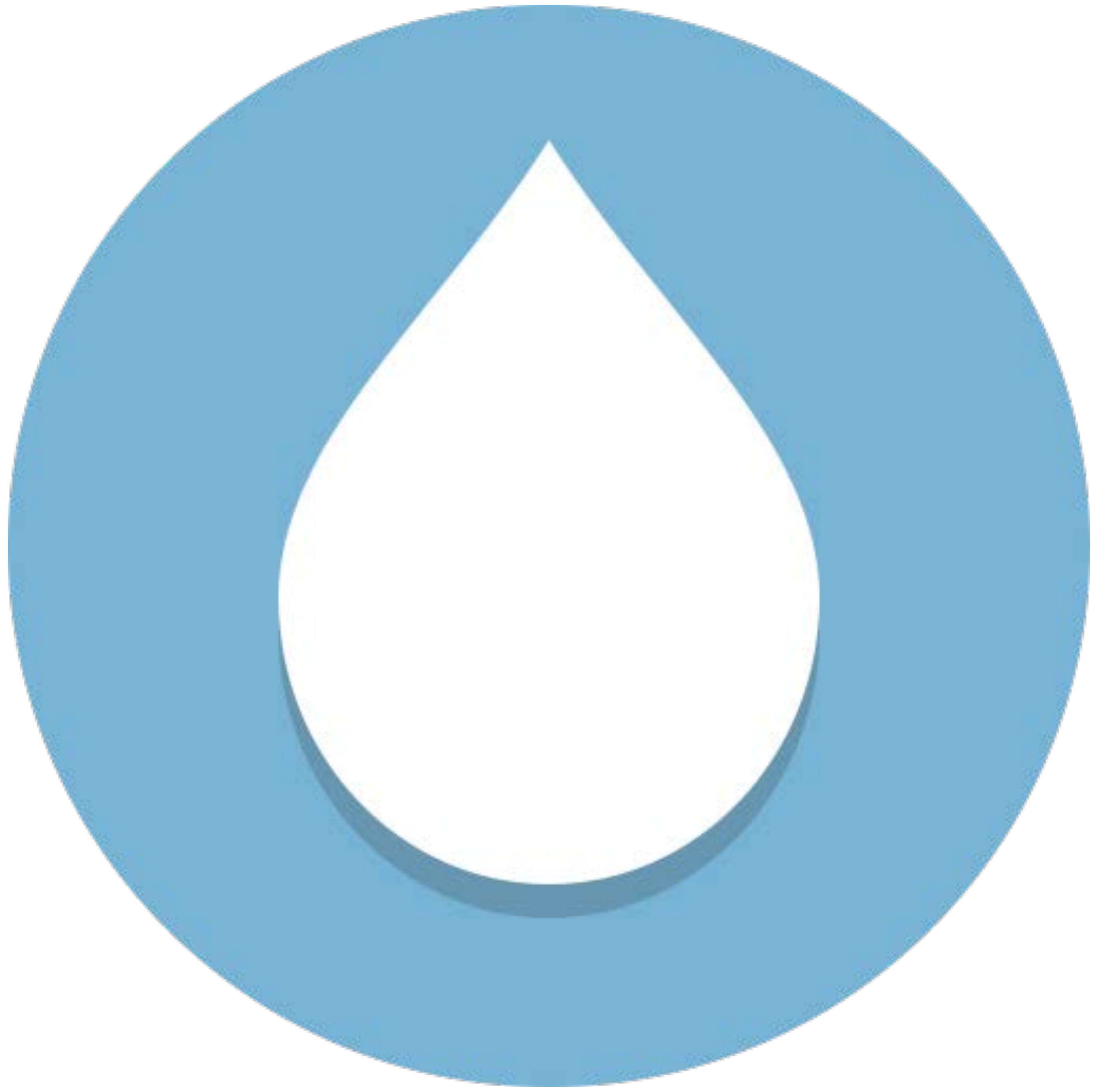
ING

hello

10 questions you ask at the start of your API

5 to zoom out

5 to zoom in



qover

English ▾

How much might I potentially lose?

After **3** year(s), I might lose **€9,749** on the purchase price of my vehicle in case of a total loss

I want to be covered for €10,000 ▾

○
○
○
●

You chose the product GAP Insurance - formula 3 years

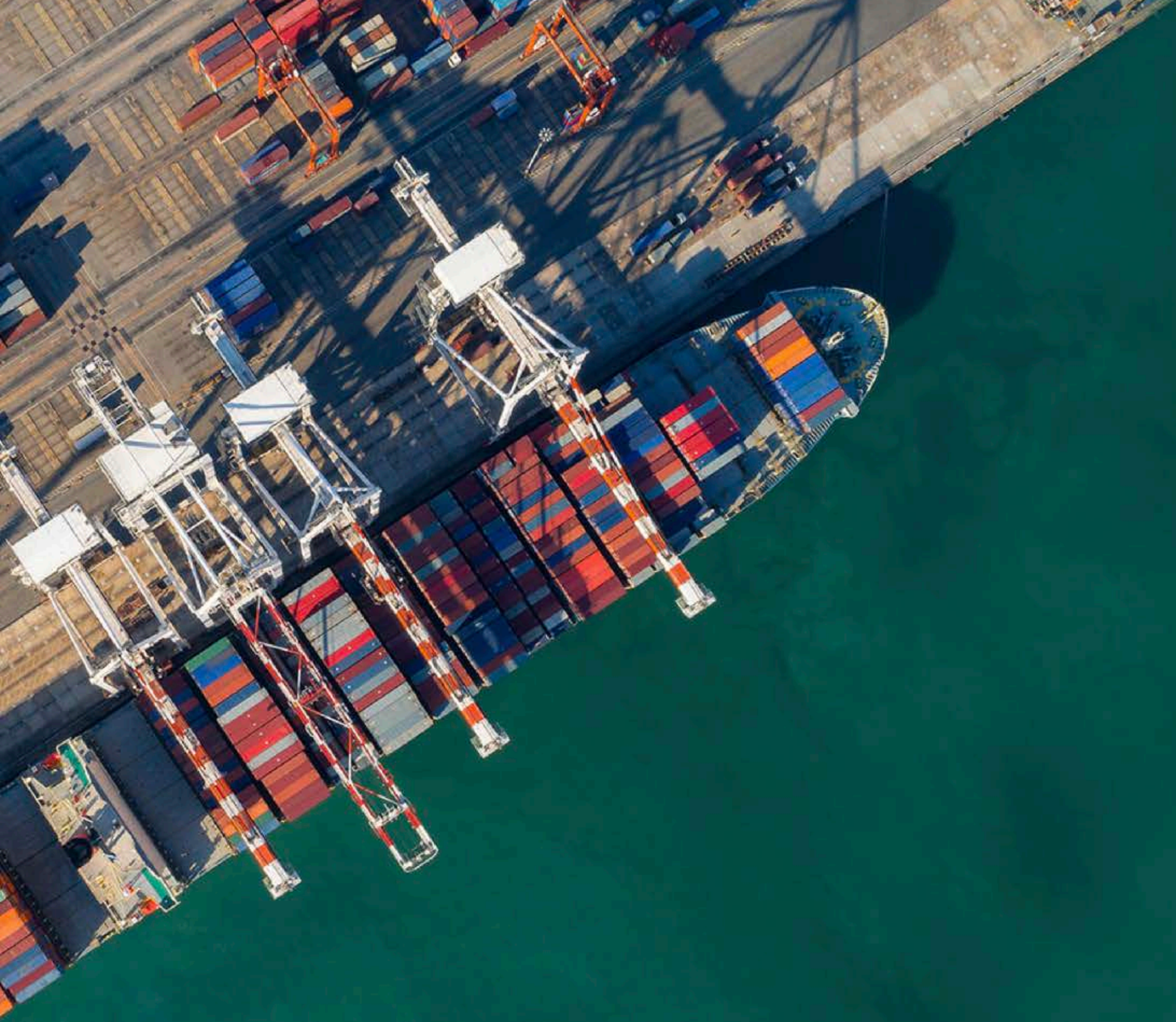
Buy now: €205.92 / year

Want a 10% discount?
Contact your broker or your car dealer.

MY CAR

ABARTH 500	1	**€23,232.00**	Start Date: 2017-05-03

POTENTIAL LOSS WITHOUT GAP
€9,749
POTENTIAL CASCO PAYOUT
€13,483
1
€92.70
2
€344.70
3
Total price: €617.76



Capture Your Shipment's Full Journey

Seamless Integration

Custom ID fields enable seamless integration with your existing SCM or ERP systems.

Monitor Shipments In-Transit

View active shipments' current status, location, and triggered alerts at any time.

Access to Historical Data

A complete log of all your past shipments enables detailed analysis.

Robin
Conference Room
Available
Next event at 4:00 pm
Start Meeting
3:43pm
TODAY
May 29, 2015
Mighty Meeting
[illegible]

0

';--have i been pwned?

Check if you have an account that has been compromised in a data breach

Generate secure, unique passwords for every account

email address
pwned?
Learn more at 1Password.com

10 questions you ask at the start of your API

#1

WHAT IS OUR DESIGN CHALLENGE?

problem

ultimate impact

possible solutions

context & constraints

How might we,
help Dutch and Belgium retail customers
to easily split their restaurant bill and request
payments from friends - so that the hassle of money
doesn’t interfere with a good evening out?

#2

WHAT ARE OUR TOP-LEVEL RESOURCES?

/details

/items

..[or]..

/appointments

/credit-cards/application

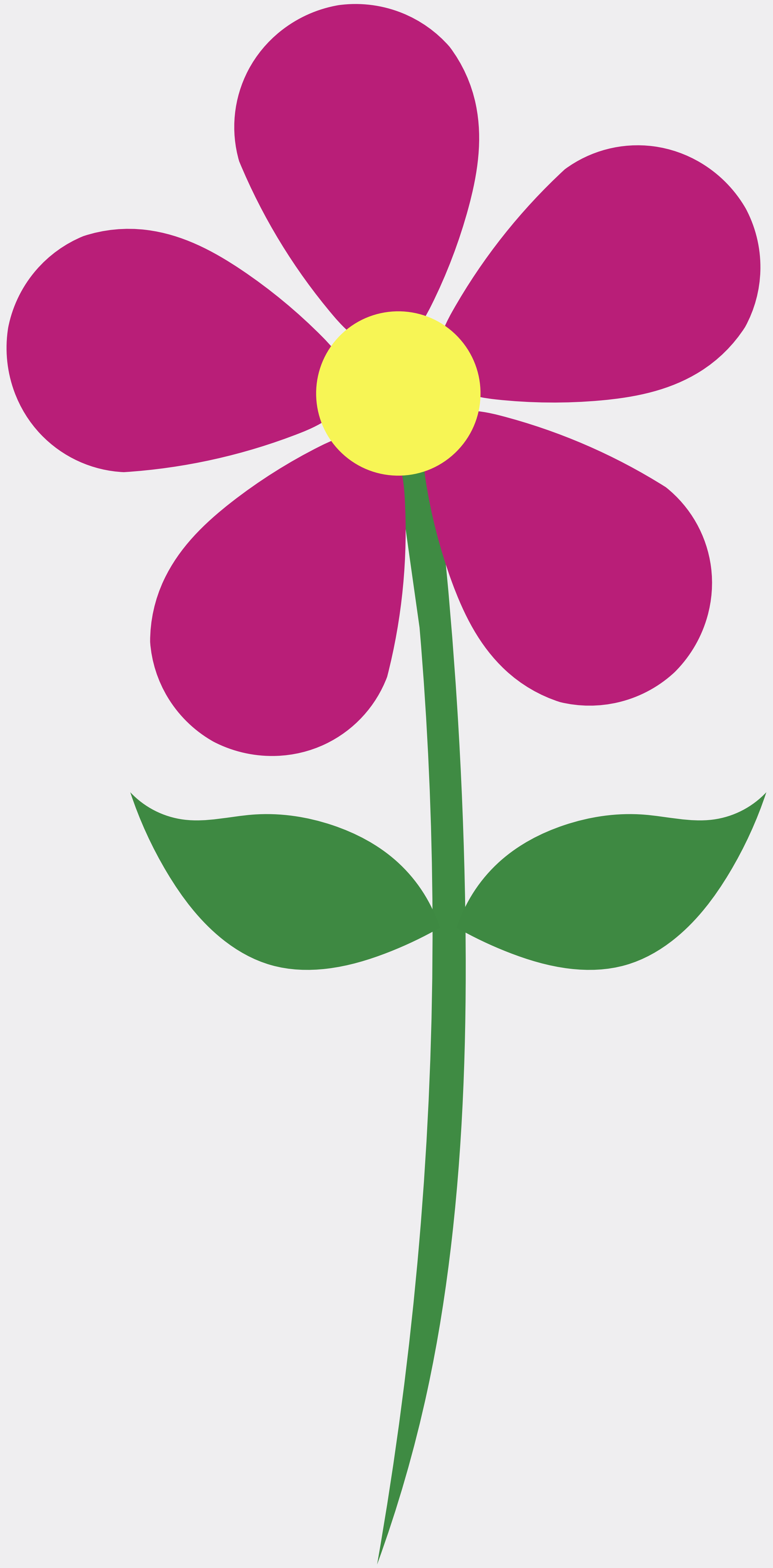
/current-accounts/transactions

“There are only two hard things in Computer Science: cache invalidation and naming things.”

–Phil Karlton

#3

IS OUR API COHESIVE?



#4

WHAT PRINCIPLES, POLICIES & VALUES DO WE CODIFY?

May 25th 2018

GDPR

#5

HOW DO WE MONETIZE OUR API?

FREE, PAY-AS-YOU-GO, TIERED PRICING, FREEMIUM, UNIT-BASED PRICING, TRANSACTION FEE, REVENUE SHARING, COST-PER-CLICK, COST-PER-ACTION, RECURRING REVENUE STREAMS, UP-SELL OPPORTUNITY, INTERNAL USAGE AND MANY OTHERS

#6

WHICH DESIGN PERSPECTIVE DO WE USE?

screen

stage

self

screen

viewpoint of single application

often User Interface driven, guards a specific experience

some internal steps of the process are folded

enforce behavior for different applications

validation and messages match with the front end

```
$ post https://api.ing.nl/mobile/credit-card/requests
{name, telephone, email, home address}
> 201 Created

$ post https://api.ing.nl/mobile/credit-card/requests/{id}/step1
{monthly income, annual income}
> 200 OK

$ post https://api.ing.nl/mobile/credit-card/requests/{id}/step2
{requested limit, card type, payment options}
> 200 OK

$ post https://api.ing.nl/mobile/credit-card/requests/{id}/step3
> 200 OK
```

stage

viewpoint of capturing the essence

maximum flexibility, enables multiple experiences

comprises of logical units you update

little to no notion of time

can be used with multiple interaction patterns

```
$ post https://api.ing.nl/credit-card/requests
> 201 Created

$ patch https://api.ing.nl/credit-card/requests/{id}
{name, telephone, email, home address}
> 200 OK

$ patch https://api.ing.nl/credit-card/requests/{id}
{monthly income, requested limit, card type, payment options}
> 200 OK

..[repeat]..

$ post https://api.ing.nl/credit-card/requests/{id}
> 201 Created
```

self

viewpoint of internal processes

exposes various states and intermediate steps

behaves CRUD like or chatty

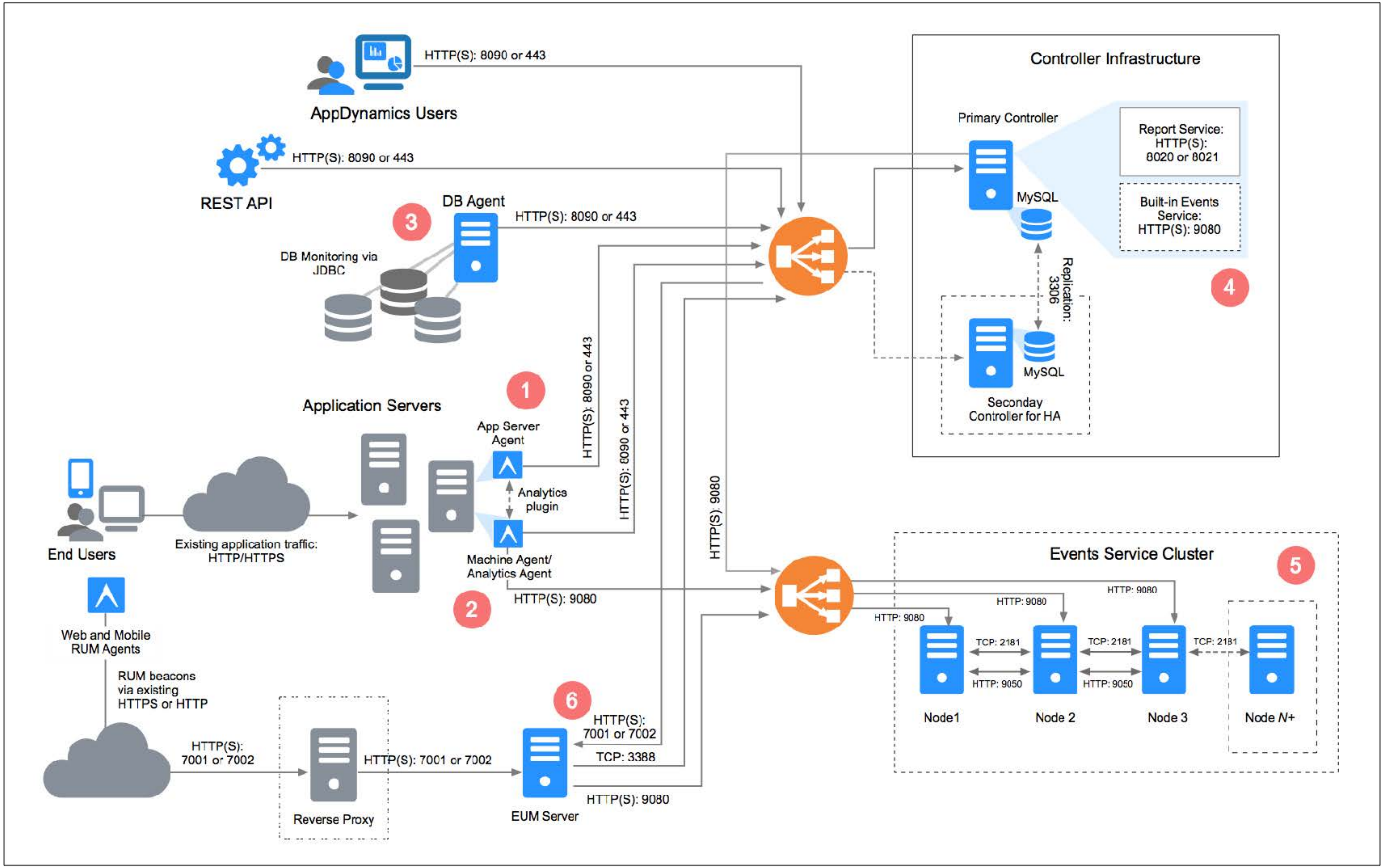
references system specifics, often with magic numbers

offers great level of control

```
$ post https://api.ing.nl/credit-card/requests
> 201 Created

$ post https://api.ing.nl/credit-card/requests/{id}/bkr-toets
> 200 OK

$ post https://api.ing.nl/credit-card/requests/{id}/check-income
> 200 OK

$ post https://api.ing.nl/credit-card/requests/{id}/assess-risk
> 200 OK

$ post https://api.ing.nl/credit-card/requests/{id}/issue-card
> 200 OK

$ post https://api.ing.nl/credit-card/requests/{id}/send-letter
> 200 OK
```

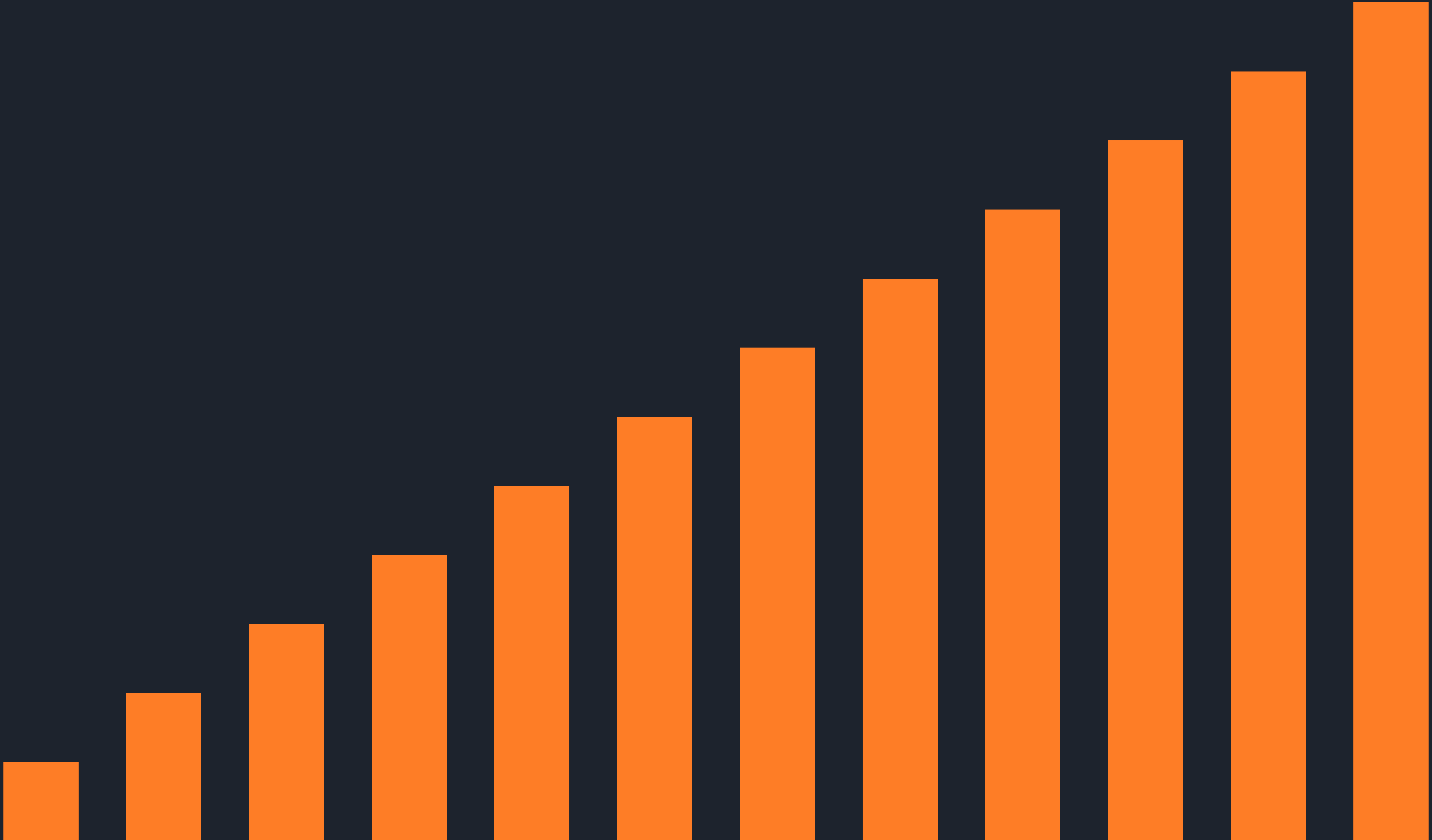
#7

WHAT SYSTEMS DO WE NEED FOR OUR API?


AppDynamics Users
HTTP(S): 8090 or 443
REST API
HTTP(S): 8090 or 443
DB Agent
3
HTTP(S): 8090 or 443
DB Monitoring via JDBC
Controller Infrastructure
Primary Controller
Report Service: HTTP(S): 8020 or 8021
Built-in Events Service: HTTP(S): 9080
MySQL
Replication: 3306
4
MySQL
Seconday Controller for HA
Application Servers
1
App Server Agent
HTTP(S): 8090 or 443
HTTP(S): 8090 or 443
HTTP(S): 9080
Analytics plugin
Machine Agent/ Analytics Agent
2
HTTP(S): 9080
End Users
Existing application traffic: HTTP/HTTPS
Web and Mobile RUM Agents
RUM beacons via existing HTTPS or HTTP
Events Service Cluster
5
HTTP: 9080
HTTP: 9080
HTTP: 9080
TCP: 2181
TCP: 2181
TCP: 2181
HTTP: 9050
HTTP: 9050
Node1
Node 2
Node 3
Node N+
HTTP(S): 7001 or 7002
Reverse Proxy
HTTP(S): 7001 or 7002
6
HTTP(S): 7001 or 7002
TCP: 3388
HTTP(S): 9080
EUM Server

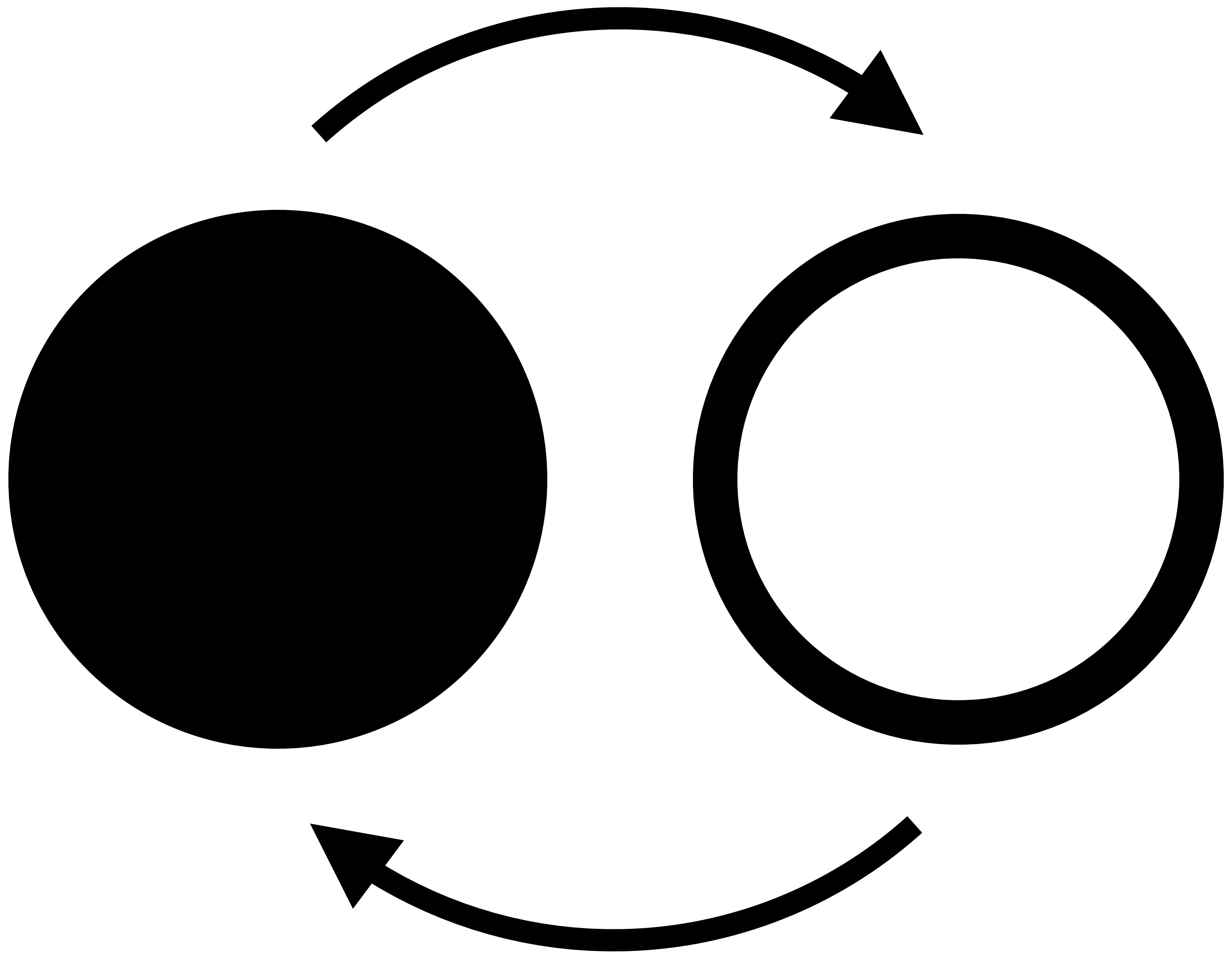
#8

WHAT NEEDS TO SCALE, AND BY HOW MUCH?



#9

HOW DO WE EVOLVE OUR API?



#10

HOW DO WE MEASURE OUR ULTIMATE IMPACT?

```
$ cat ./api-start-deck.md l grep #

#1  what is our design challenge
#2  what are our top-level resources?
#3  is our API cohesive?
#4  what principles, policies & values do we codify?
#5  how do we monetize our API?
#6  which design perspective do we use?
#7  what systems do we need for our API?
#8  what needs to scale, and by how much?
#9  how do we evolve our API?
#10 how do we measure our ultimate impact?

$ _
```

Good APIs don’t happen by accident