clojure@runa

runa

a somewhat non-technical talk

a tale of two e-tailers

a (really) large online auction site

runa

5000 requests/sec

1 ms response times

runa

5 ms with network

1 (+9) machines

averaging 30 million calls a day

up to 500 million calls a day

single developer

nearly 100% Clojure

thanks, Clojure!

1.5% site coverage

a (cool) billion dollars of offers a day

early days, 5% lift

runa

tale #2

large online shoes retailer

~40% coverage

18% lift

Whee!

behind the scenes

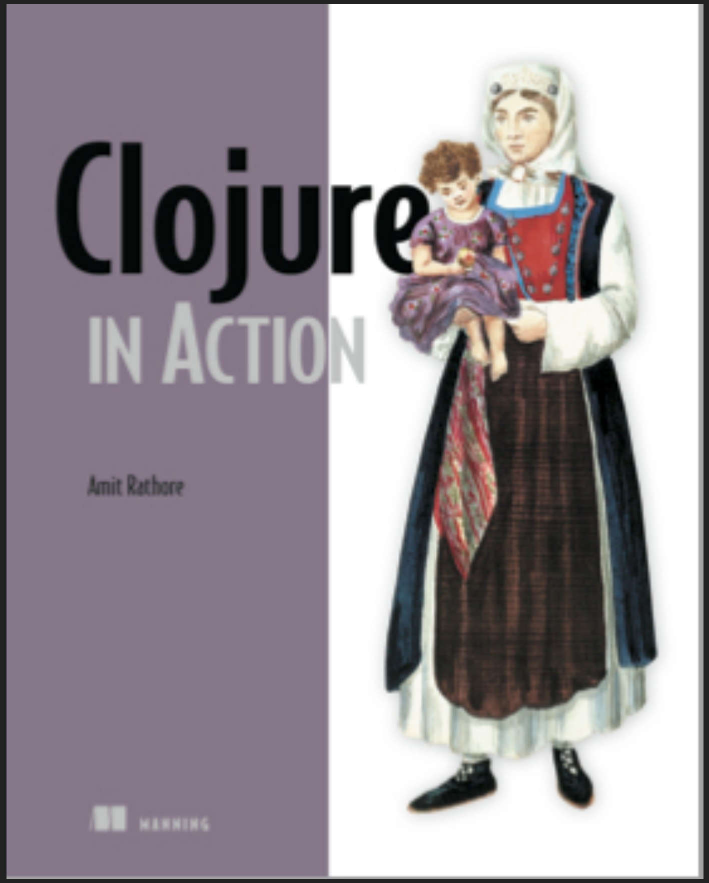
DSLs for fun and profit

for dynamic pricing

instant personal deals

Amit Rathore

VP of Engineering


Clojure
IN ACTION
Amit Rathore
MANNING

DSLs

many talks

this talk

not how

but why

not for developers

but for non-developers

business people

yes, s-expressions

what this talk is really about

empowering non-technical folks

What is Runa?

Runa starts where Google stops

eCommerce 2.0

SaaS

click-stream

SKU-level

user behavior

big data

machine learning

statistical models

predict purchase intent

Smart Deals

instant personal deals

right shopper

right product

right offer

right time

artificial intelligence

math reborn

decision trees

support vector machines

agent-based modeling
logistic regression
hidden-markov models
hinged planes
...

more sale dollars

less discount dollars

bottom line impact

more sales, profitably

big data
+ machine learning
+ predictive modeling
+ instant personal deals
= **profitable sales lift**

runa

DSLs

business rules

margin management

promotions management

free shipping perimeters

traffic segmentation

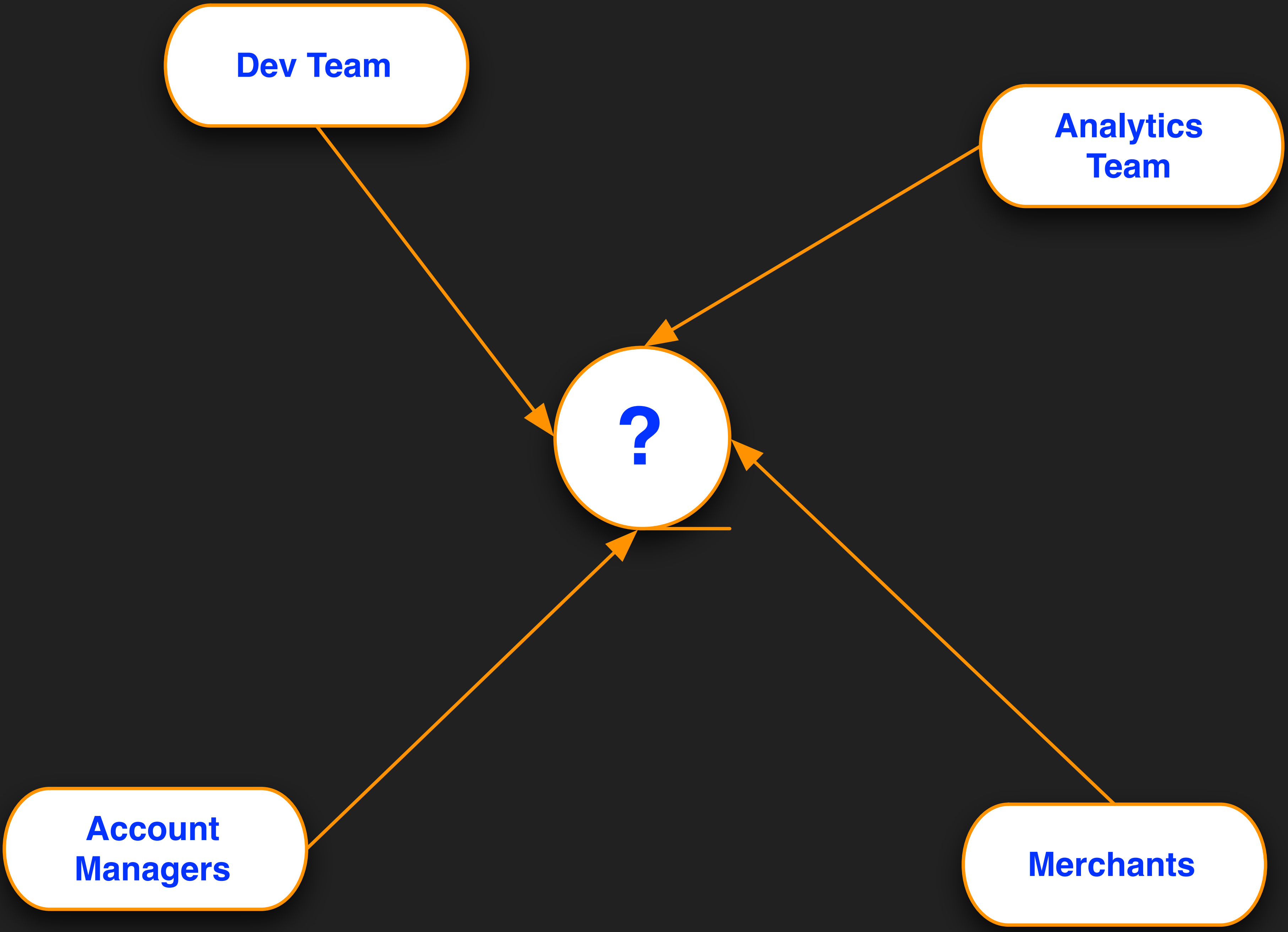
rules management

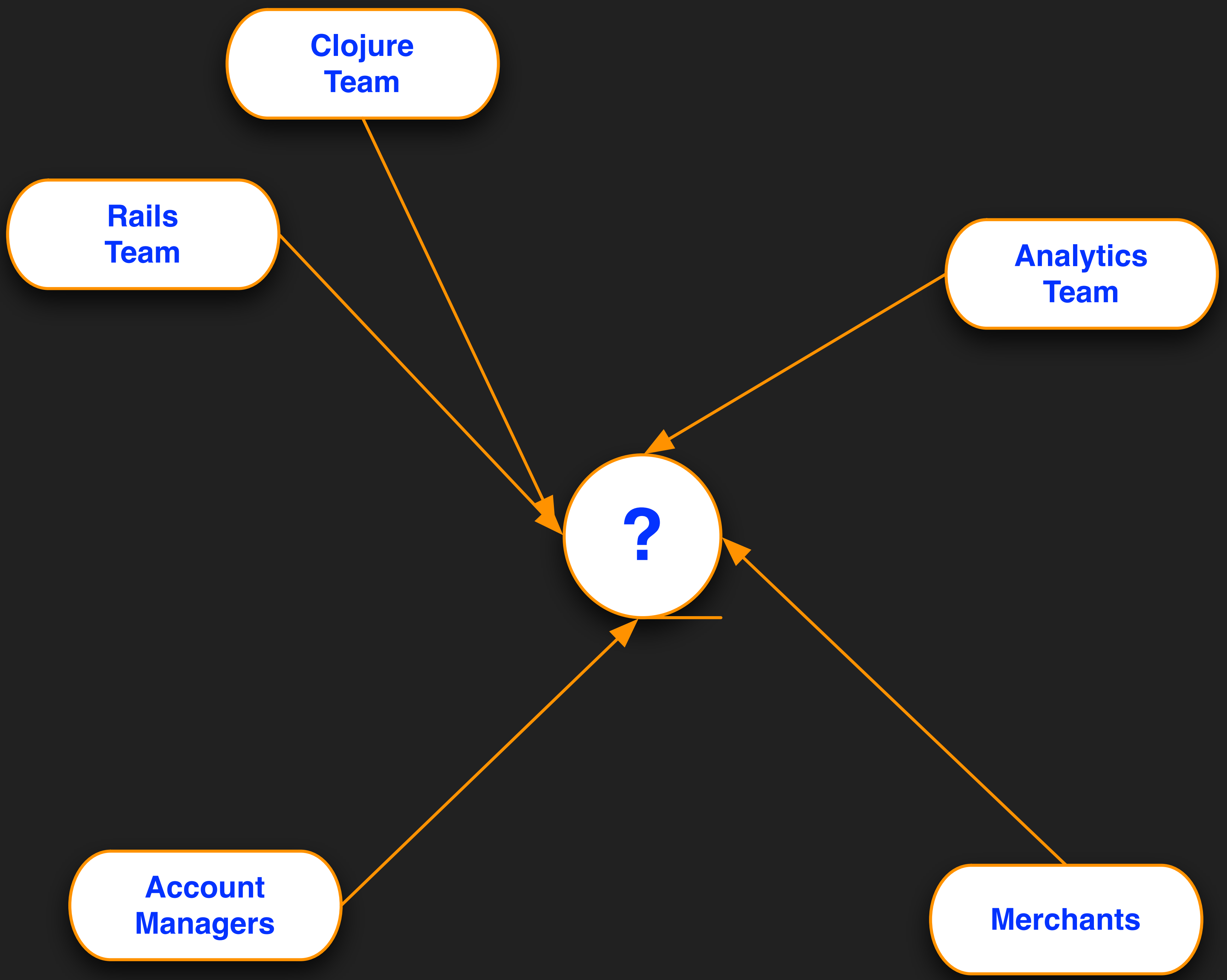
account managers

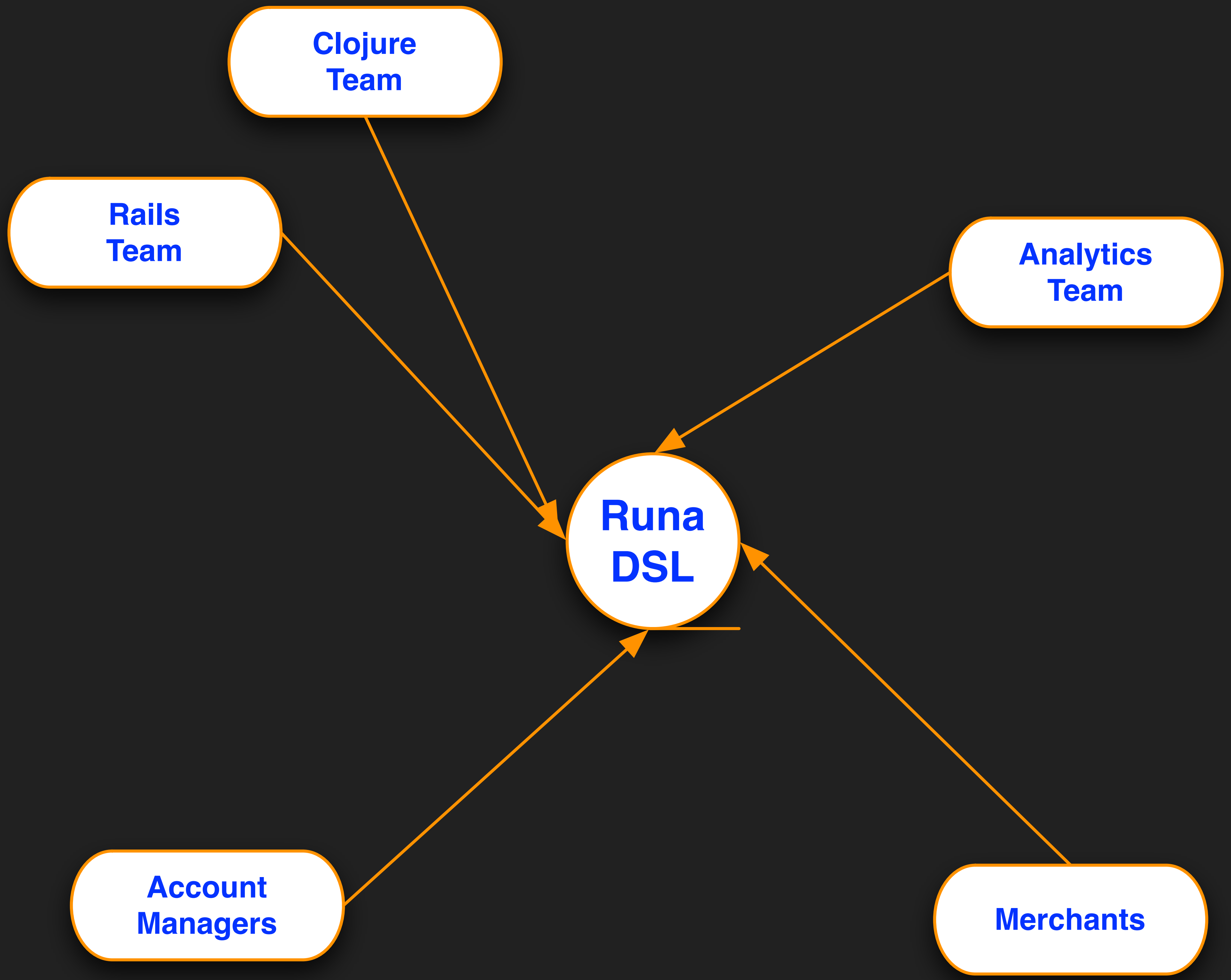
analytics team

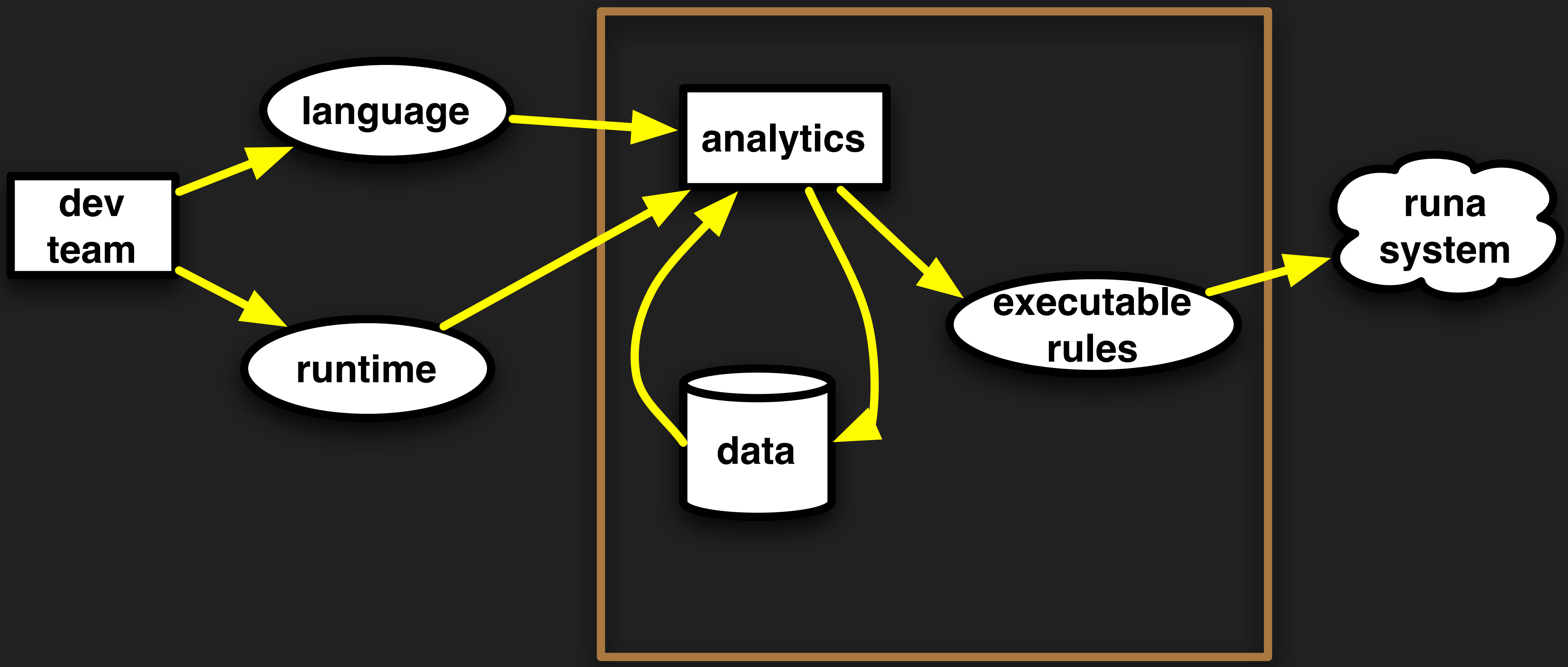
merchants

loosely coupled

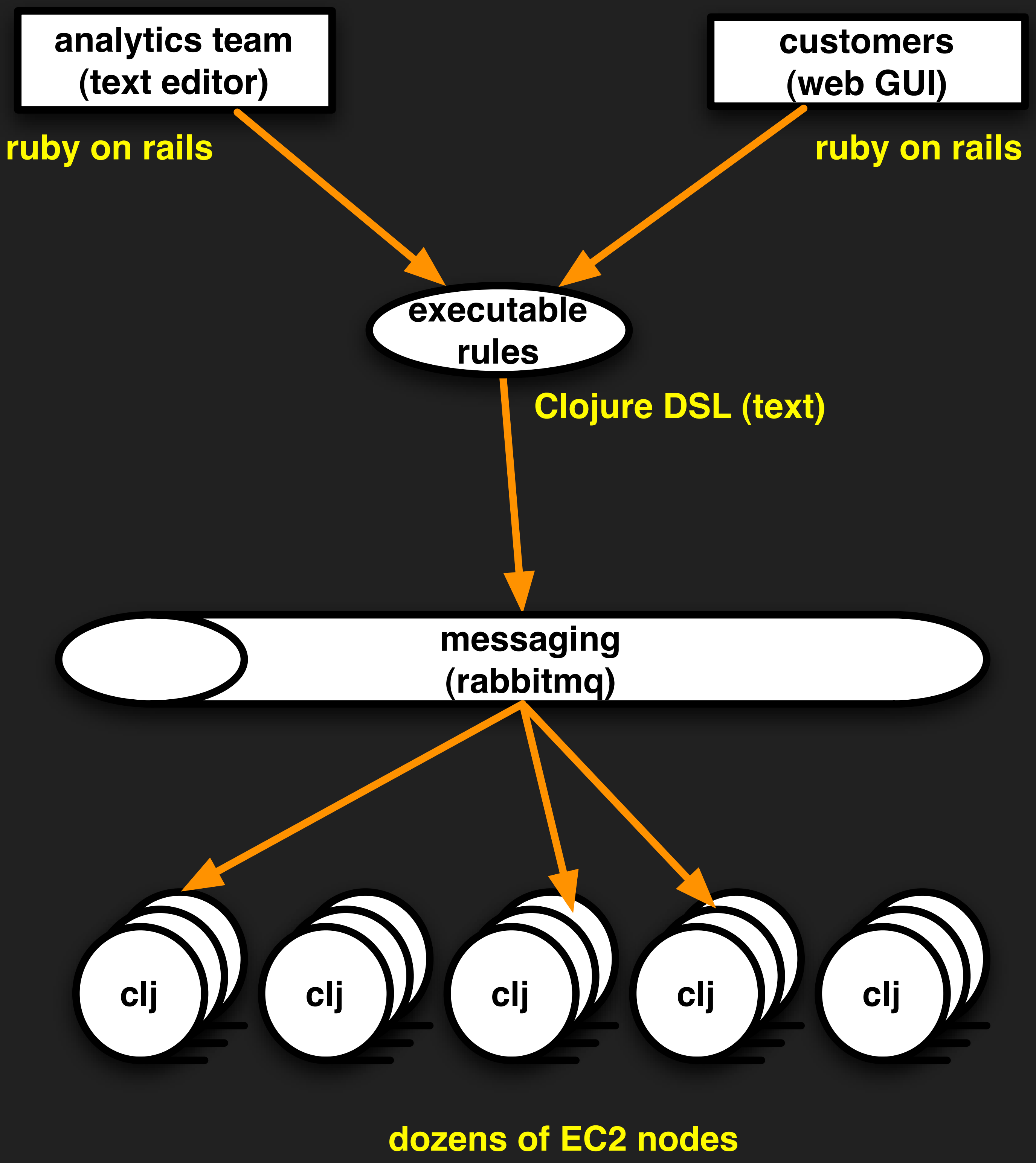

Dev Team
Analytics Team
?
Account Managers
Merchants


Clojure Team
Rails Team
Analytics Team
?
Account Managers
Merchants


Clojure Team
Rails Team
Analytics Team
Runa DSL
Account Managers
Merchants



no dev/deployment needed


analytics team (text editor)
ruby on rails
customers (web GUI)
ruby on rails
executable rules
Clojure DSL (text)
messaging (rabbitmq)
clj
clj
clj
clj
clj
dozens of EC2 nodes

examples

segmentation

```
(def-segment :arts-merchant seg-c
  (runa-abandonment-index 99.5)
  (criteria
   (and
    (< $time-to-first-cart 30)
    (contains? $search-terms "masters"))))
```

```
(def-segment :carving-blocks seg-a
  (runa-abandonment-index 98.5)
  (criteria
   (and
     (empty? $url-referrer)
     (> $purchases:all:amount 400))))
```

```
(def-segment :carving-blocks seg-a
  (runa-abandonment-index 98.5)
  (criteria
   (and
     (empty? $url-referrer)
     (> $runa-purchases:30-days:count 10))))
```

promotions

```
(def-special-promo :shoes-merchant coupon7
  (type :coupon)
  (scope :item)
  (sticky true)
  (code "RCOM7")
  (criteria
    (not (matches? $product-name ["ugg"])))
  (description "Runa Coupon")
  (cost
     (amount 7)
     (denomination "%" )))
```

```
(def-promo :mangotree-hotels free-breakfast
  (type :coupon)
  (code "FBF100")
  (description "Free Breakfast")
  (overlay-lightbox-text "Have breakfast on us!")
    (cost
     (amount 15)
     (denomination :money)
     (frequency :daily)))
```

runa

exclusions

```
(consumer-filters :mangotree-hotels
  (ip-addresses "10.1.4.5" "10.1.4.16")
  (ip-address-range "10.1.4.0 - 10.1.4.255")
  (referrer-containing "nextag.com"))
```

deal delivery

```
(delivery-rule :carving-blocks blanket-rule
  (segments
   (none-of? :DirectTraffic :SearchTraffic))
  (pages *)
  (methods *))
```

```
(delivery-rule :rambo-inc recapture-rule
  (segments
    (not seg-c))
  (pages
   (not :home :detail))
  (methods
   (or :pre-abandonment :recapture)))
```

UI customization

```
(def-template-rule :shoes-merchant pre-ui
  (templates :ugg-pre-abandonment)
  (segments *)
  (methods :pre-abandonment)
  (pages *)
  (promo-type :coupon)
  (map-status *)
  (criteria
    (empty? (sticky-incentive)))
  (promos :coupon-ugg))
```

controlling features

```
(feature-bit :baseline-sampling true)

(feature-bit :pre-abandonment-2g-ui true)

(feature-bit :back-button
  (is-not-in? $merchant-id ["arts-merchant" "shoes-merchant"]))
```

systems integration

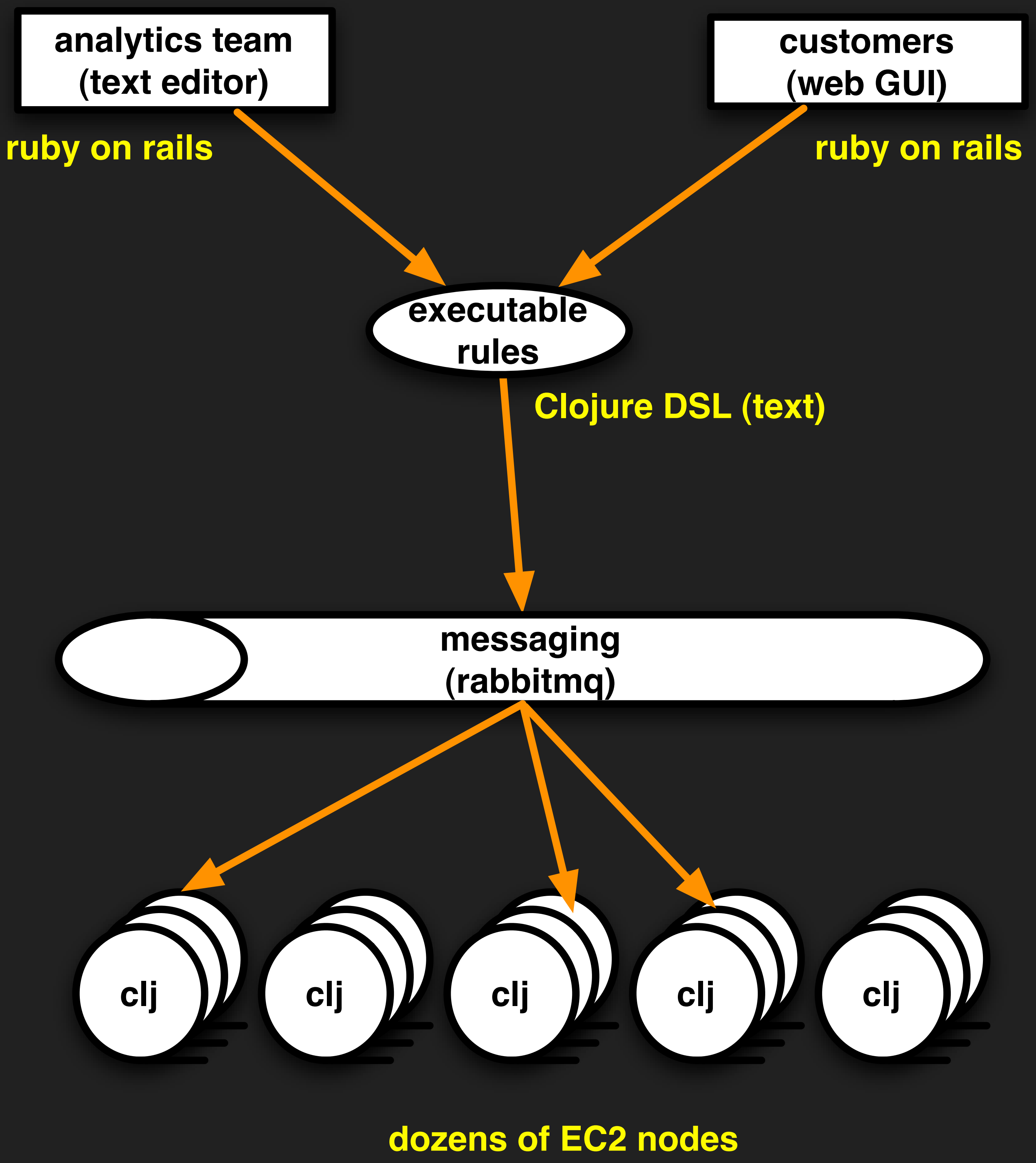
Ruby on Rails -> Clojure

code-generation

Rails-side

runa

benefits


analytics team
(text editor)
customers
(web GUI)
ruby on rails
ruby on rails
executable rules
Clojure DSL (text)
messaging
(rabbitmq)
clj
clj
clj
clj
clj
dozens of EC2 nodes

high-level language

version control

rule change vs. impact

bottom-up

domain-driven design

formal specifications

distributed teams

concerns

parenthesis and normals

live code updates

syntactic checks

semantic checks

trial runs

live-code updates

Alan J. Perlis

“I think that it's extraordinarily important that we in computer science keep fun in computing. When it started out, it was an awful lot of fun. Of course, the paying customers got shafted every now and then, and after a while we began to take their complaints seriously. We began to feel as if we really were responsible for the successful, error-free perfect use of these machines. I don't think we are. I think we're responsible for stretching them, setting them off in new directions, and keeping fun in the house...”

runa

?

amit@runa.com