crushing the monolith

mats cronqvist, may 2013

but I like monoliths...

alt text

"Klarna's next generation architecture:
A soft-realtime, distributed, no-master system enabling the best SLA in online payments."

`id`

- Ph.D. in experimental Nuclear Physics
  - hardware
  - online software
  - offline software
- 10 years @ ericsson
  - building
  - testing
  - troubleshooting
  - supporting
- 5 years @ klarna
  - developer
  - architect
- github.com/massemanet

klarna vision

Safer

Simpler

More fun

klarna goals

keep it simple for the consumer

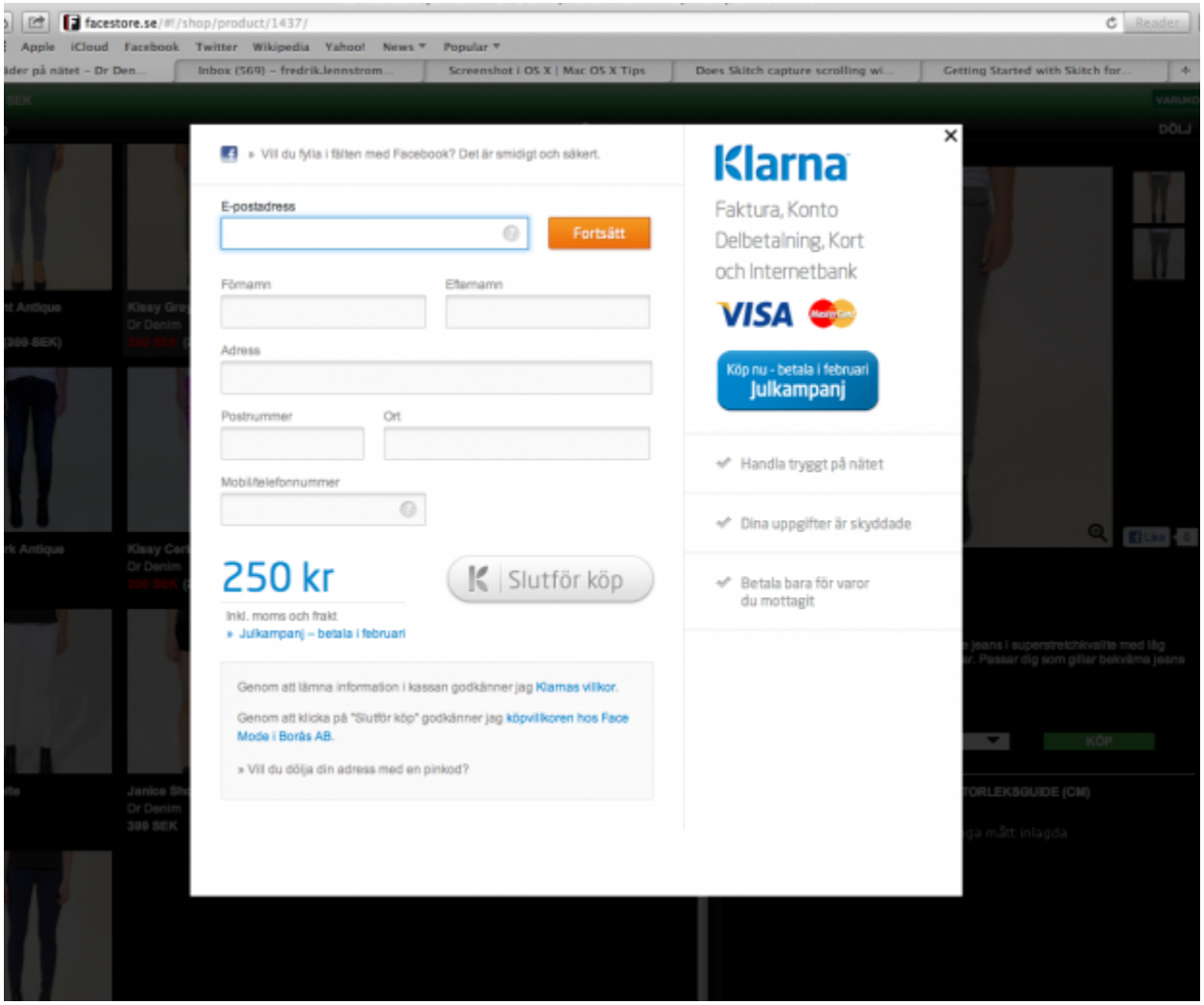
- no pre-registration
- id with top of mind info
- separate buying from paying

we'll want to buy more stuff on the internet it the stores send us stuff, and we only pay the store if we like the stuff.

klarna requirements

- invoicing
  - with real-time risk assessment (~3s)
  - with unreliable data
- many settlement options
- customer care (~500)
- fraud
  - merchants
  - consumers
- very high availability (~99.9)

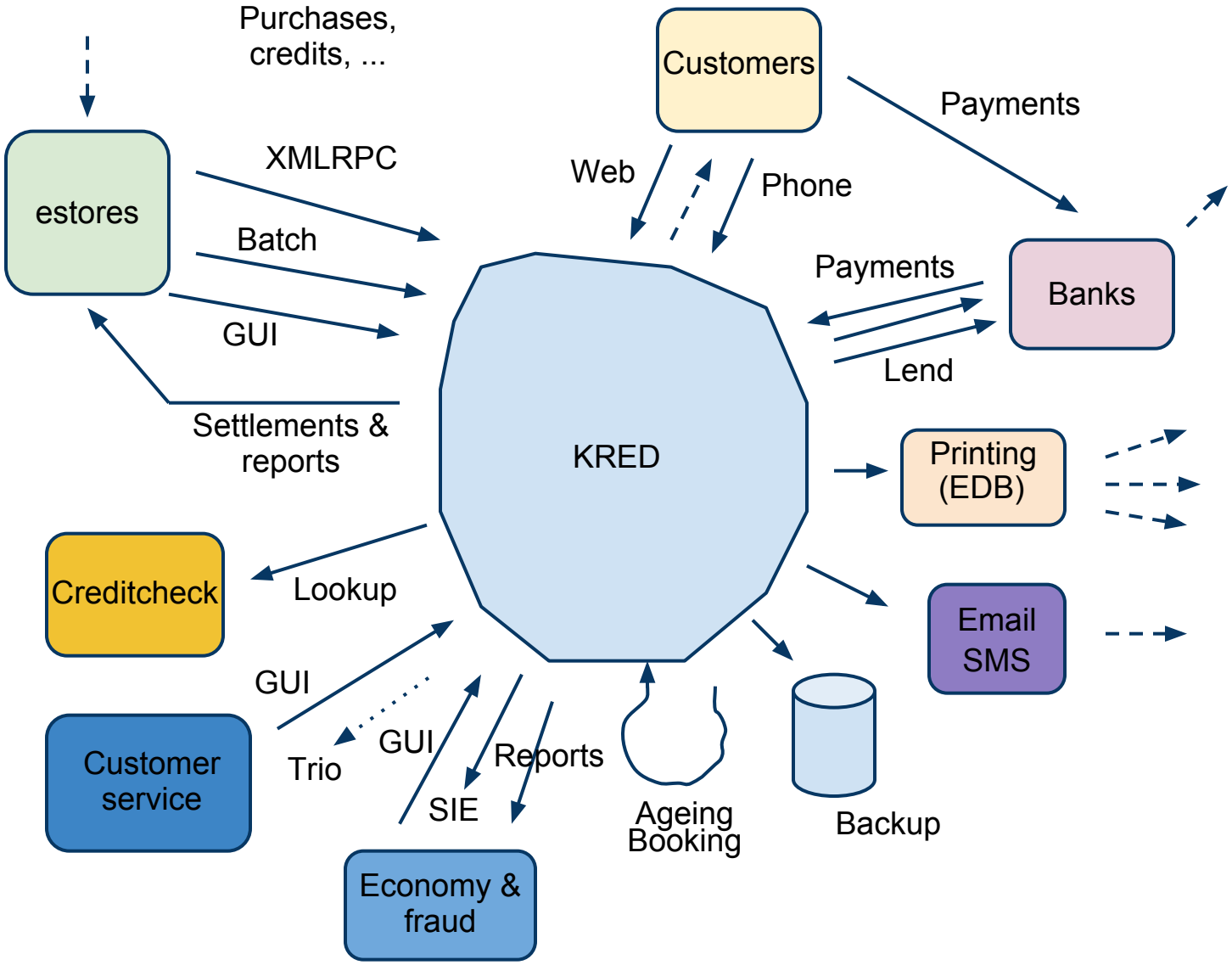
checkout


facestore.se/#!/shop/product/1437/
» Vill du fylla i fälten med Facebook? Det är smidigt och säkert.
E-postadress
Fortsätt
Förnamn
Efternamn
Adress
Postnummer
Ort
Mobiltelefonnummer
250 kr
Inkl. moms och frakt
» Julkampanj – betala i februari
Slutför köp
Genom att lämna information i kassan godkänner jag Klarnas villkor.
Genom att klicka på "Slutför köp" godkänner jag köpvillkoren hos Face Mode i Borås AB.
» Vill du dölja din adress med en pinkod?
Klarna
Faktura, Konto
Delbetalning, Kort
och Internetbank
VISA
Köp nu - betala i februari
Julkampanj
Handla tryggt på nätet
Dina uppgifter är skyddade
Betala bara för varor
du mottagit

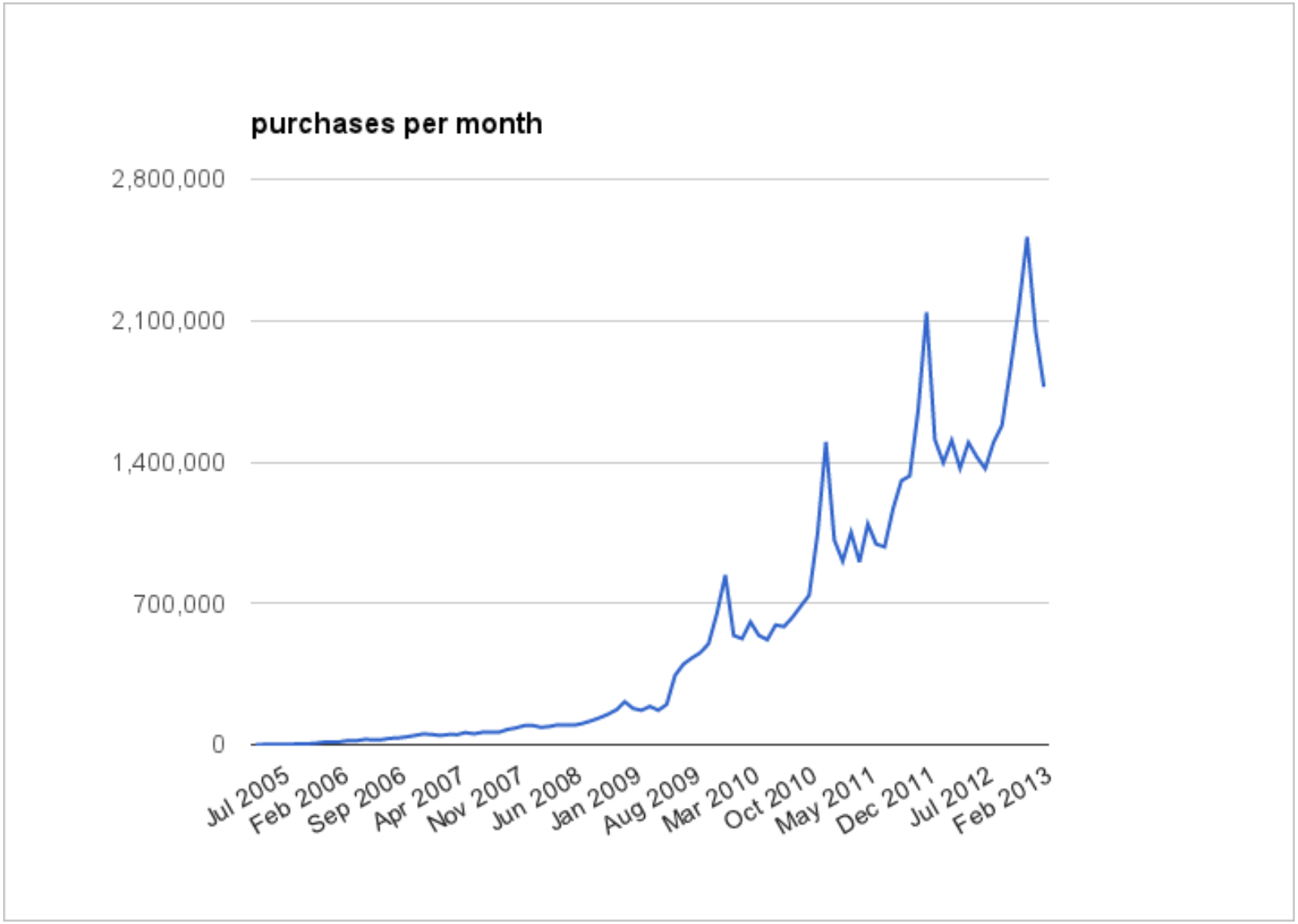
`secret sauce`

- order creation API
- ID service
  - identify a legal entity
  - on the internet...
- risk service
  - prevent fraud and defaults
- in "real time"...

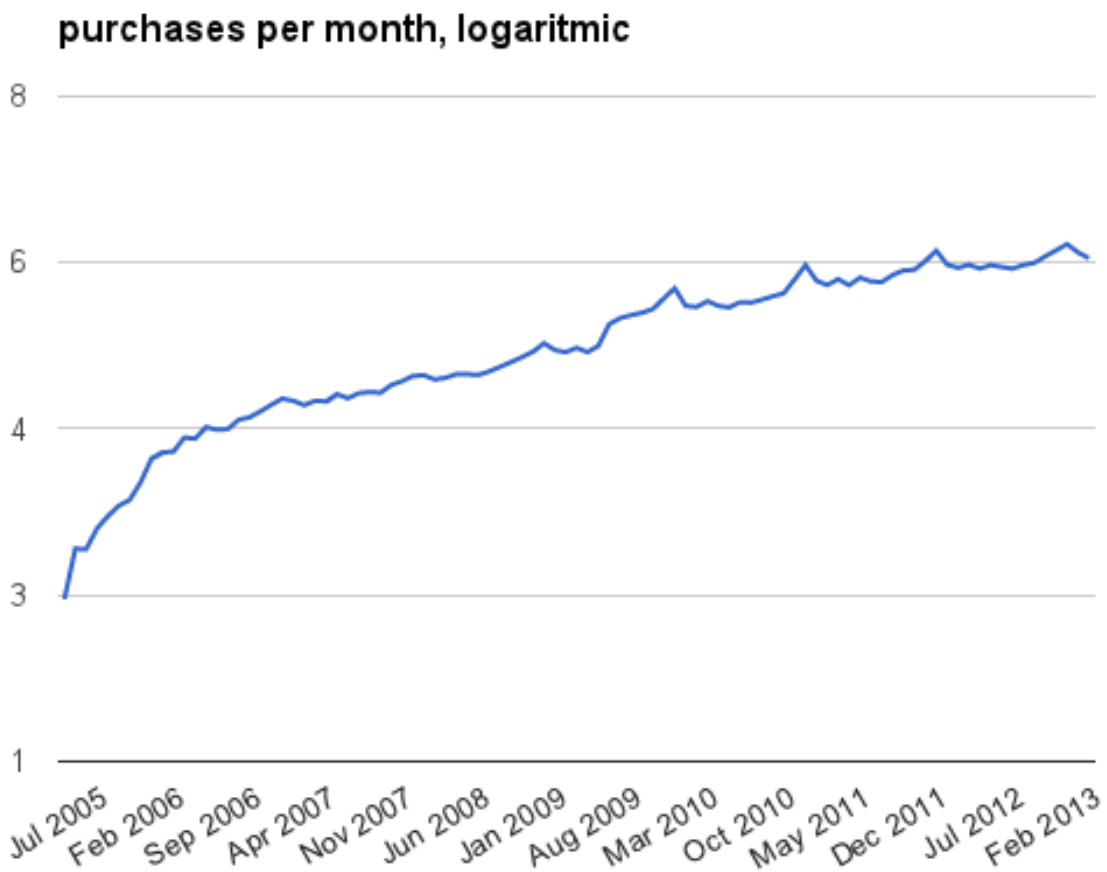
kred


Purchases,
credits, ...
Customers
Payments
estores
XMLRPC
Web
Phone
Batch
Payments
Banks
GUI
Lend
Settlements &
reports
KRED
Printing
(EDB)
Creditcheck
Lookup
Email
SMS
GUI
Customer
service
Trio
GUI
Reports
SIE
Ageing
Booking
Backup
Economy &
fraud

growing pains



magnitudes

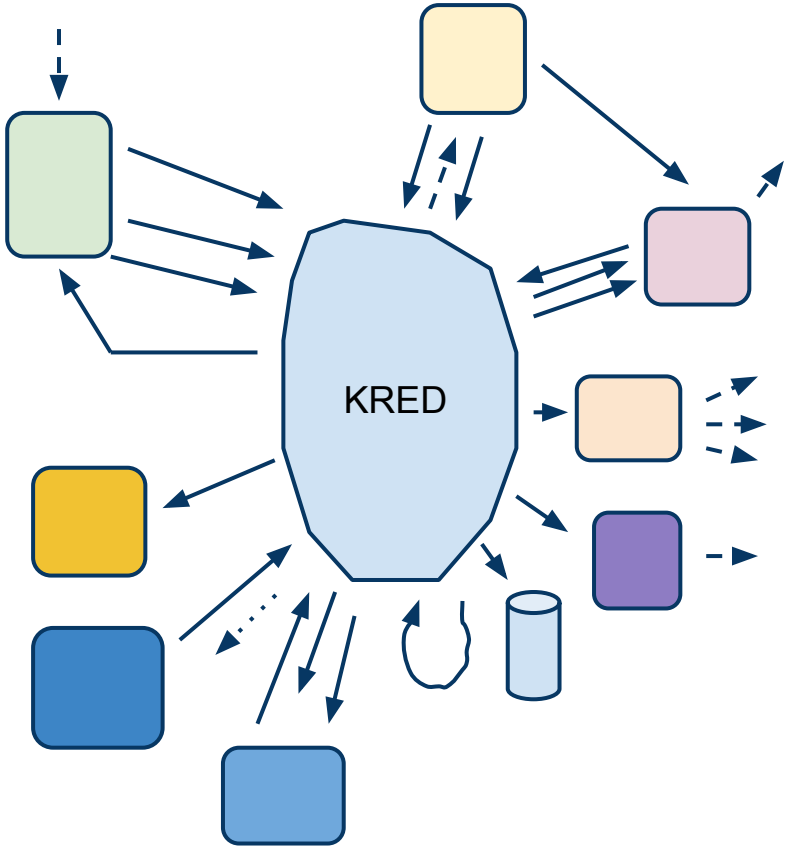

purchases per month, logaritmic
8
6
4
3
1
Jul 2005
Feb 2006
Sep 2006
Apr 2007
Nov 2007
Jun 2008
Jan 2009
Aug 2009
Mar 2010
Oct 2010
May 2011
Dec 2011
Jul 2012
Feb 2013

dtop

```
--------------------------------------------------------------------------------
kred@kred-psize: 493.4G(530.6G), cpu%: 362(366), procs: 1.6k, runq: 0, 16:17:03
memory:      proc   18.4G, atom    5.0M, bin  496.1M, code   57.0M,  ets  474.4G

pid            name                          current              msgq    mem cpu
<0.28894.66>   yaws_server:acceptor0/2        etrans:mk_date_li       0   17.0G 190
<0.25.0>       file_server_2                 gen_server:loop/6       0    8.2M  11
<0.17983.74>   yaws_server:acceptor0/2        gen:do_call/4           0    1.9M   9
<0.9604.73>    yaws_server:acceptor0/2        prim_inet:recv0/3       0    7.6M   1
<0.23590.74>   yaws_server:acceptor0/2        prim_inet:recv0/3       0    1.8M   1
<0.26.0>       code_server                   code_server:loop/       0 426.0k   1
--------------------------------------------------------------------------------
```

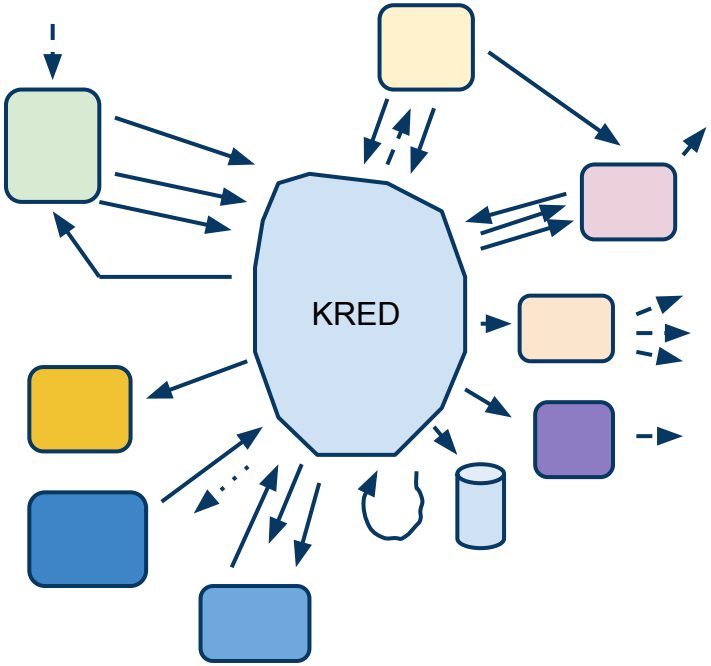
domains

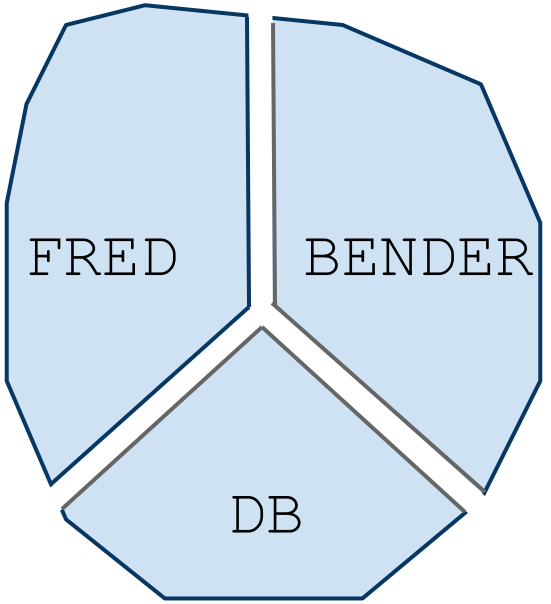


- Soft real time
- Off-line analysis
- Customer GUI
- External Systems
- Data Warehousing
- Routine financial

K2

- Break `kred` into services
  - `FRED` is one such service
  - `BENDER` is all other services

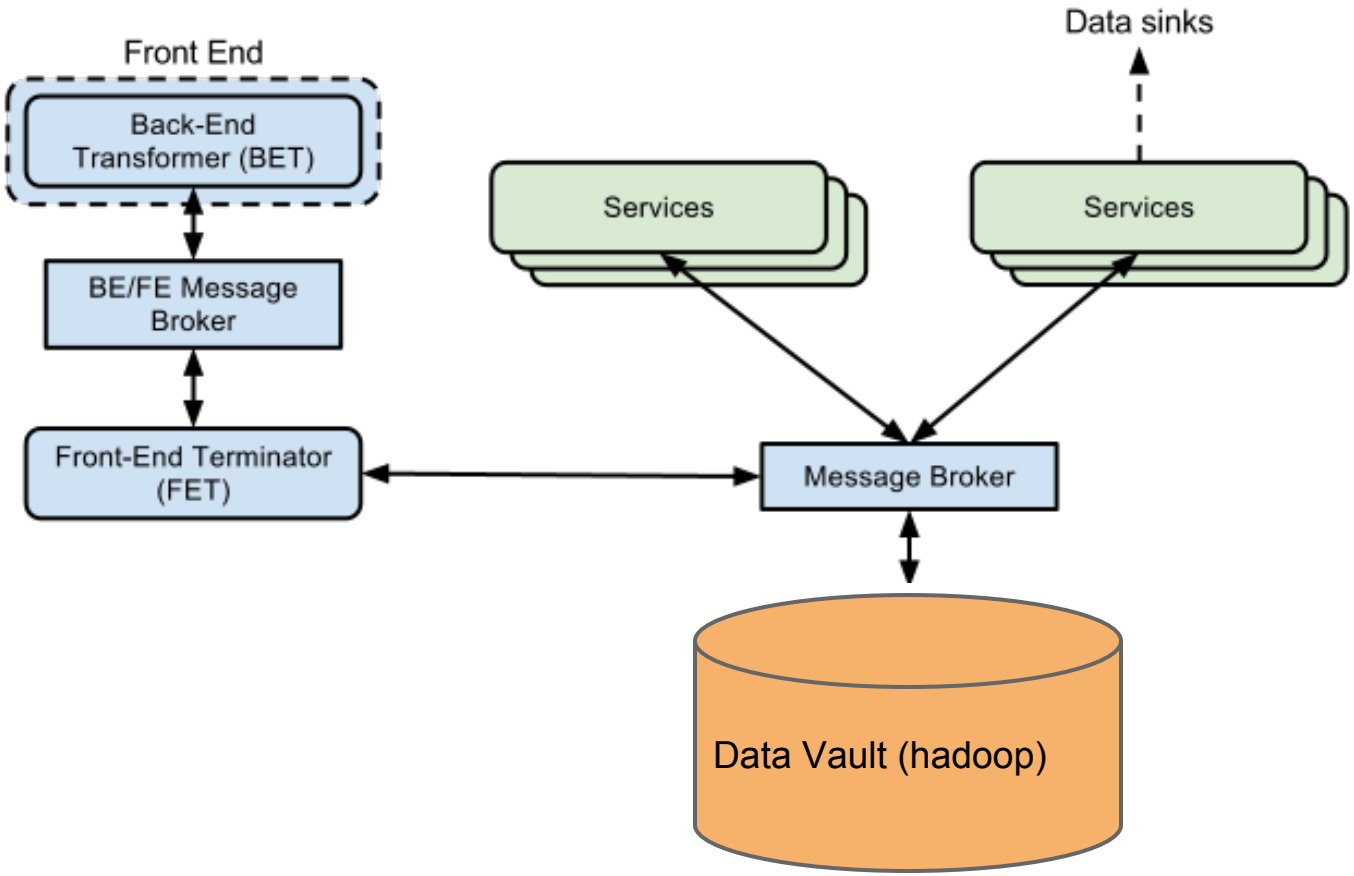




`decoupling`

- Horizontal scaling
- Concurrent and independent development
- Loose coupling, both with respect to functionality and data

BENDER from outer space



BENDER services

- GUIs
- bookkeeping
- business intelligence
- accounts
- printing
- dunning
- etc...

FRED

- Service that creates orders
  - Business logic ported from kred
  - Soft real-time (~3 seconds)
-

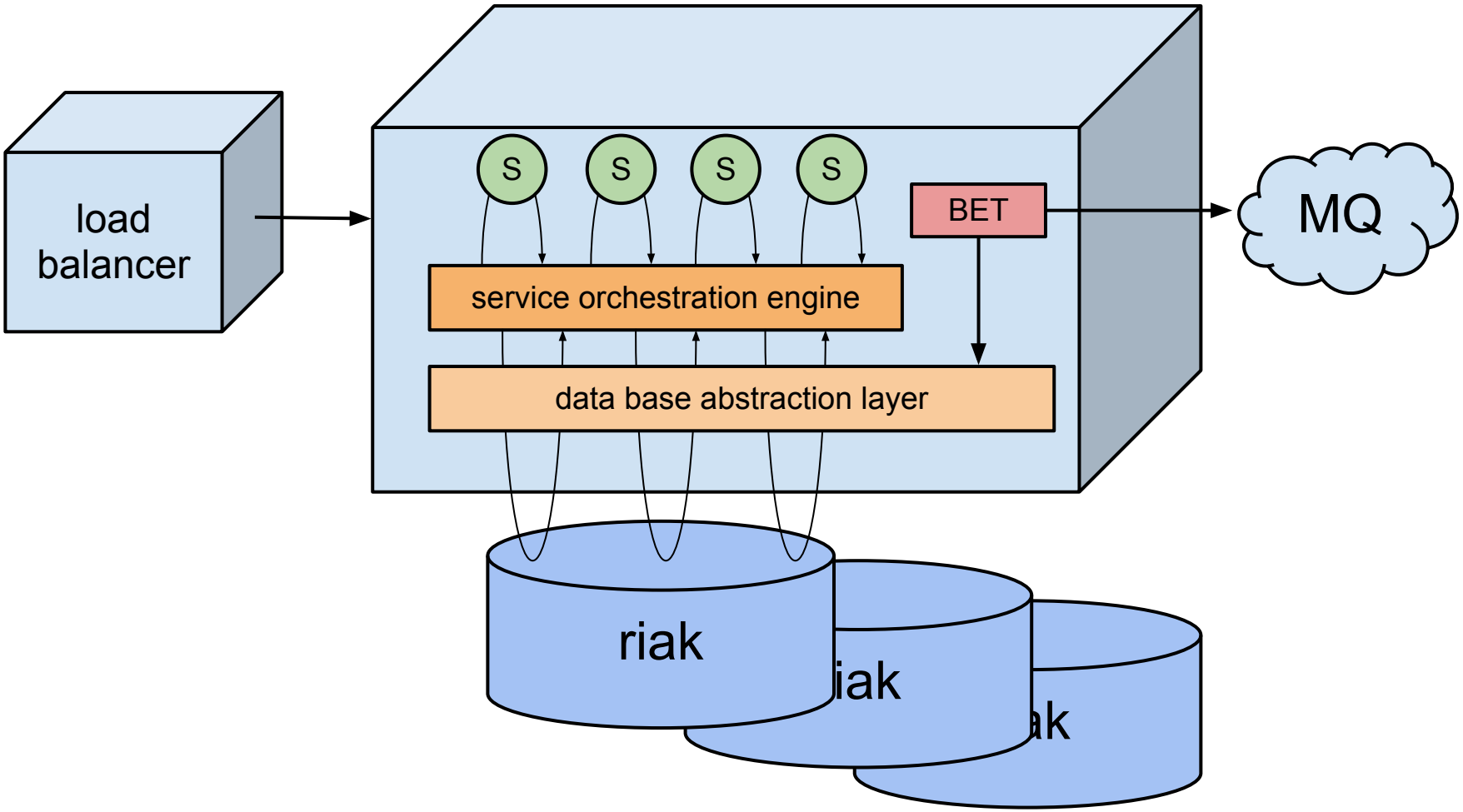
FRED scope

- some API methods
- checkout
- ID service
- risk service
  - on the spot reject (FRED)
  - external (kred)
  - fallback policy (FRED)

FRED knockouts

- Site
  - a data center
- FRED clusters
  - independent of each other
  - initially one cluster per site
- FRED machine
  - ~5 per cluster
- FRED service
  - an orchestrated component

FRED from low earth orbit



FRED tech stack

- Redhat Enterprise Linux
- webmachine
- mochiweb
- Riak
- Rabbit MQ
- Erlang
- Git
- Chef
- rebar
- splunk

why riak?

- Distribution -> no transactions
- FRED data model -> key/value
- lots of attention to operations
- basho produces quality stuff
- riak search interesting
- competition looked worse
- data center replication intriguing

siblings

- "everything that touches Riak needs to be resolvable in some way."
  - merging
  - pick one
  - make siblings impossible

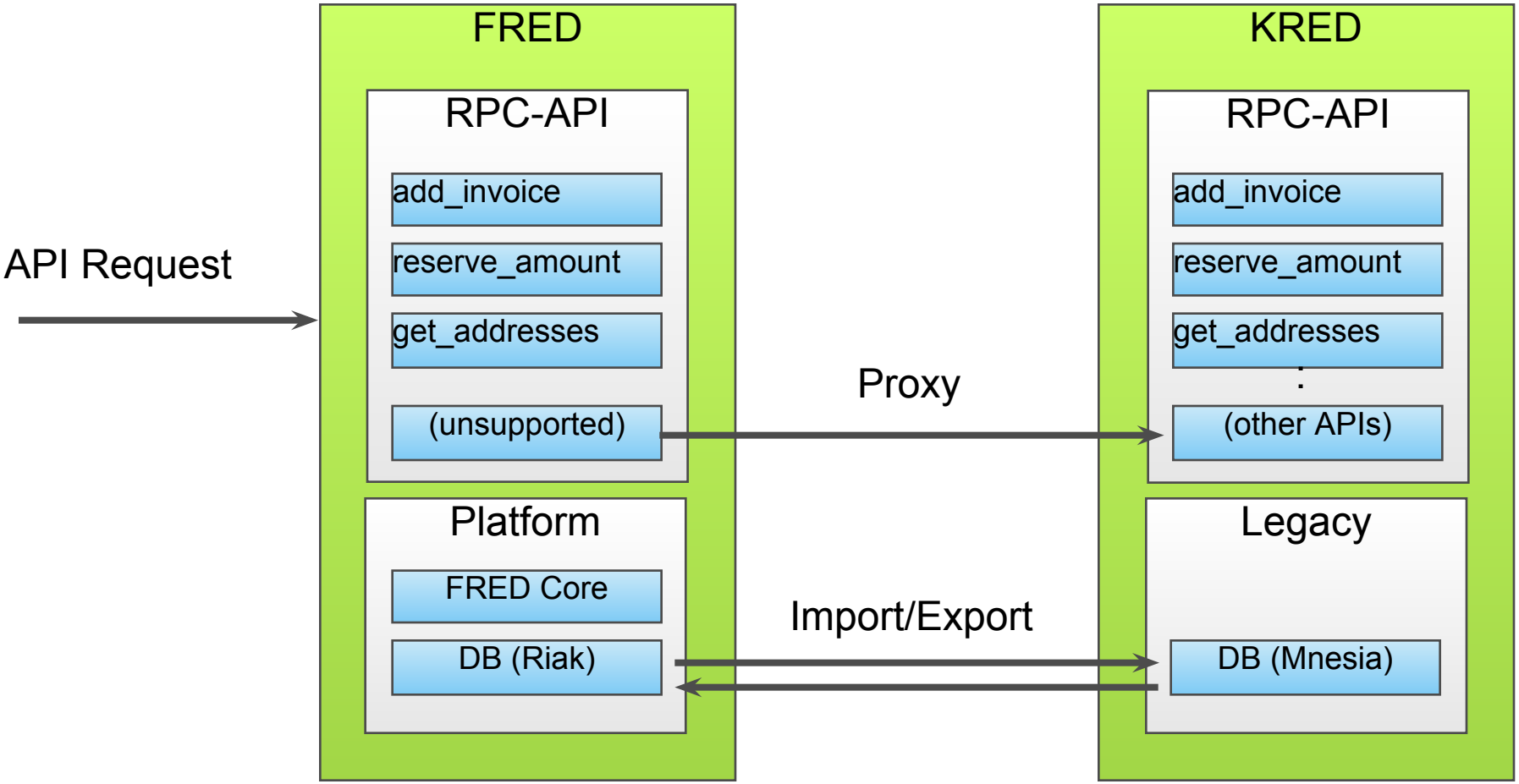
migration

- ~15,000 existing merchants
  - we must preserve (most) integrations
- ~10 versions of the API in use
- many merchants rely on quirks

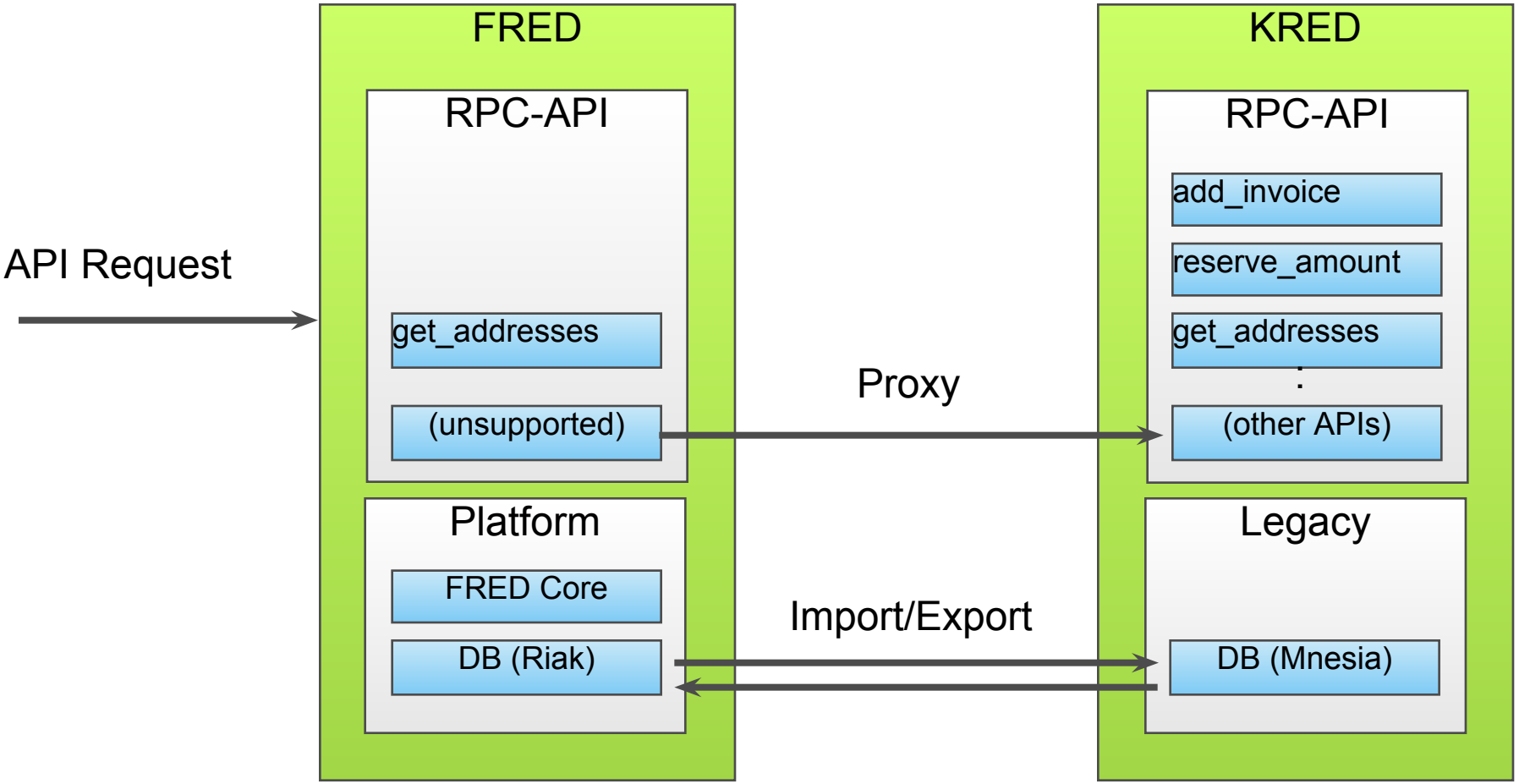
therefore

large parts of `FRED` are backported

FRED migration



status


FRED
RPC-API
get_addresses
(unsupported)
Platform
FRED Core
DB (Riak)
KRED
RPC-API
add_invoice
reserve_amount
get_addresses
(other APIs)
Legacy
DB (Mnesia)
API Request
Proxy
Import/Export

~~resistance~~

obstacles

- ~~NIH~~
- new programming model
- untangling spaghetti is hard
- lot of infrastructure, HW and SW
- compatibility
- no spec

hell yeah

A soft-realtime, distributed, no-master system enabling the best SLA in online payments.