The First Corporate Card for Startups

Building a credit card with Elixir

Introduction

What’s Brex?

What's Brex

- Financial platform
- Infrastructure disguised as a product


business
BREX
mastercard

Payments 101

Payments (Theory)

BREX

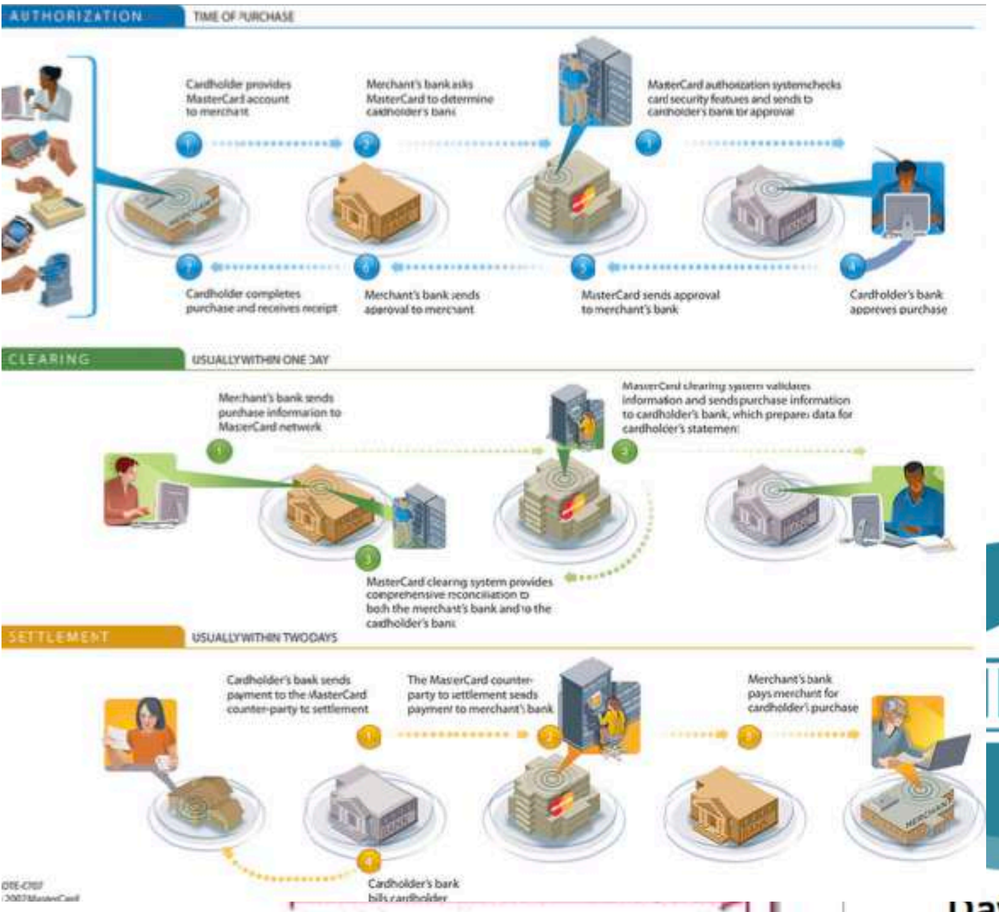
Payments are boring

BREX

~~Payments are boring~~

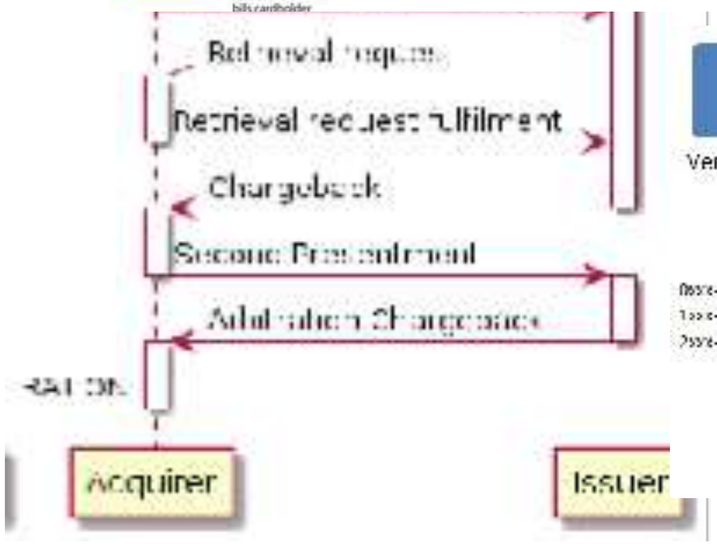
BREX

Payments are super complex

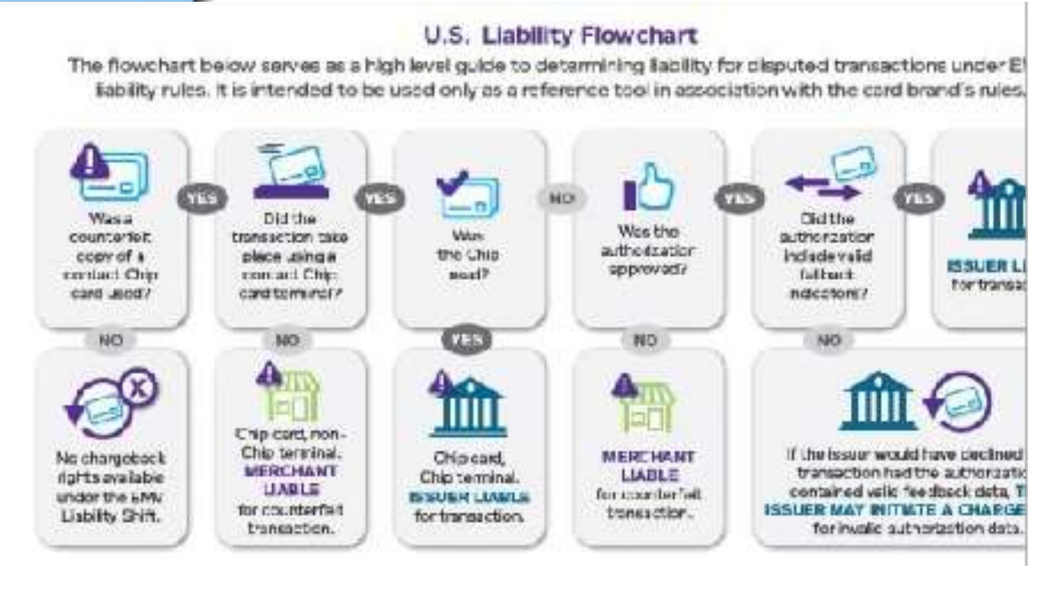

AUTHORIZATION
TIME OF PURCHASE
Cardholder provides MasterCard account to merchant
Merchant's bank asks MasterCard to determine cardholder's bank
MasterCard authorization systemchecks card security features and sends to cardholder's bank for approval
Cardholder completes purchase and receives receipt
Merchant's bank sends approval to merchant
MasterCard sends approval to merchant's bank
Cardholder's bank approves purchase
CLEARING
USUALLY WITHIN ONE DAY
Merchant's bank sends purchase information to MasterCard network
MasterCard clearing system validates information and sends purchase information to cardholder's bank, which prepares data for cardholder's statement
MasterCard clearing system provides comprehensive reconciliation to both the merchant's bank and to the cardholder's bank
SETTLEMENT
USUALLY WITHIN TWO DAYS
Cardholder's bank sends payment to the MasterCard counter-party to settlement
The MasterCard counter-party to settlement sends payment to merchant's bank
Merchant's bank pays merchant for cardholder's purchase
Cardholder's bank bills cardholder


Processing Network
VISA
MasterCard
4
5
101
011

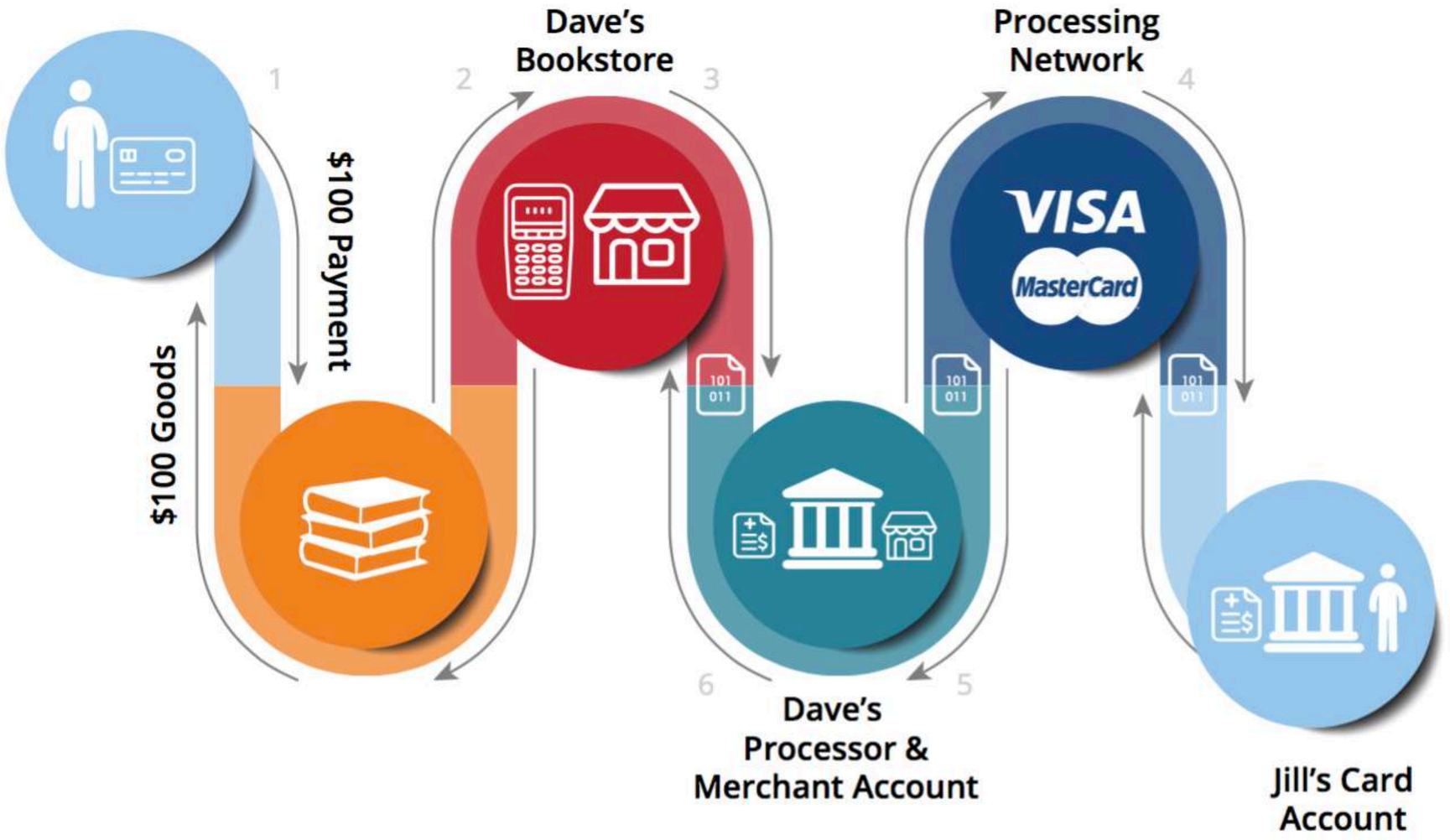

adyen
RARCLAYS
cielo
Elavon
fiserv.
Heartland
TSYS
Bank of America
CHASE Paymentech
citi
First Data
Moneris
vantiv
accel
DISCOVER
Maestro
NYCE
pulse
STAR
Interac
JCB
PLUS
UnionPay
BARCLAYS
CapitalOne
citi
HSBC
synchrony
USbank
Bank of America
Cabela's CLUB
CHASE
DISCOVER
PNC
TD Bank
USAA
Alipay
Braintree
klarna
Paysafe
Square
adyen
Authorize.Net
BlueSnap
cardconnect
PayPal
sage Pay
stripe
EVERLINK
FLAGSHIP
Paywire
PRIORITY
CAYAN
Fidelity
MERCHANT e-SOLUTIONS
National Bankcard
openedge
PIVOTAL


Retrieval request
Retrieval request fulfillment
Chargeback
Second Presentment
Arbitration Chargeback
Acquirer
Issuer


Dave's
0
Version of ISO 8583
2
Class of Message (Financial Transaction Request)
0
Function of the Message (Request)
0xxx – Authorization Message
1xxx – Financial Message
2xxx – File Action Message
4xxx – Reversal Message
5xxx – Reconciliation Message
6xxx – Administrative Message
7xxx – Fee Collection Message
8xxx – Network Management
xx0x – Request
xx1x – Request Response
xx2x – Advice
xx3x – Advice Response
xx4x – Notification
xx8x – Response Ack.
xx9x – Negative Ack.


U.S. Liability Flowchart
The flowchart below serves as a high level guide to determining liability for disputed transactions under EMV liability rules. It is intended to be used only as a reference tool in association with the card brand's rules.
Was a counterfeit copy of a contact Chip card used?
YES
Did the transaction take place using a contact Chip card terminal?
YES
Was the Chip read?
NO
Was the authorization approved?
YES
Did the authorization include valid fallback indicators?
YES
ISSUER LI
for transa
NO
No chargeback rights available under the EMV Liability Shift.
NO
Chip card, non-Chip terminal. MERCHANT LIABLE for counterfeit transaction.
YES
Chip card, Chip terminal. ISSUER LIABLE for transaction.
NO
MERCHANT LIABLE for counterfeit transaction.
NO
If the issuer would have declined transaction had the authorizatio contained valid feedback data, T ISSUER MAY INITIATE A CHARGE for invalid authorization data.

BREX


Dave's
Bookstore
Processing
Network
1
2
3
4
$100 Payment
$100 Goods
VISA
MasterCard
101
011
6
5
Dave's
Processor &
Merchant Account
Jill's Card
Account

Payment Networks

- Very old technology
- Information is scarce
- Most recent from scratch implementation is more than 10 years old

We have different constraints

BREX

Constraints

- Correctness
- Availability
- Performance

BREX

Constraints

- Correctness
- Availability
- ~~Performance~~

Performance is not that relevant

(This only applies to credit card processing)

- Average TPS worldwide is 1500/s
- Visa's maximum throughput is around 50k/s

BREX

Availability is gold

- No one likes getting a decline
- People just expect their money to be always available

BREX

Correctness keeps you alive

- Bad Information = Bad Decisions
- We’re dealing with money

The road so far

Why elixir?

BREX

Why Elixir

- Our previous company was NodeJS
- We love functional

Why Elixir

- Functional
- Battle tested runtime
- Macros are fun
- Built with distribution in mind
- Much more

What did we get wrong?

BREX

Distribution is cool...

But it doesn't solve our problem.

- Not that useful at the beginning
- Hard to interop with

Hot reloads

- Hot reloads are hard if you have state
- Not so useful in our use case

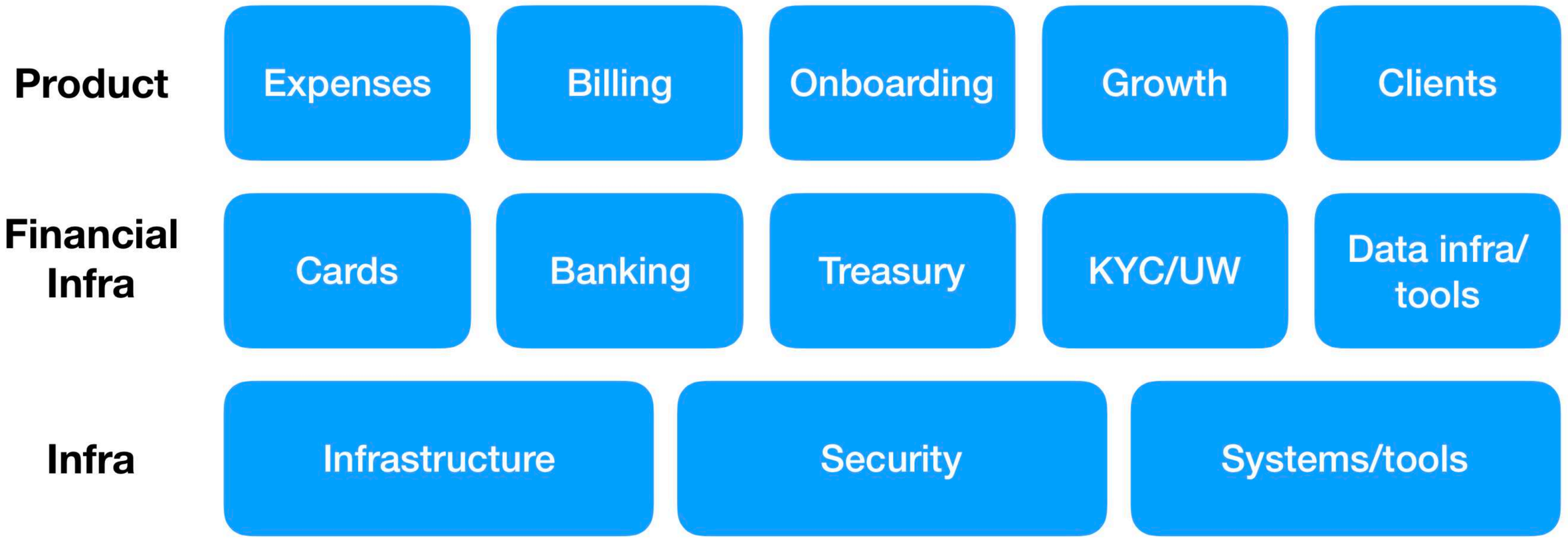
BREX

Tooling

- Works *very* well alone
- Some are hard to integrate

Architecture Overview

BREX


Product
Expenses
Billing
Onboarding
Growth
Clients
Financial Infra
Cards
Banking
Treasury
KYC/UW
Data infra/ tools
Infra
Infrastructure
Security
Systems/tools

BREX

Architecture Overview

- Service oriented
- Independent, isolated services
- Mostly stateless

BREX

Architecture Overview

- GraphQL API
- RESTful APIs
- No frontend

BREX

```
~/s/g/v/c/apps on feature/promocard ⇄ ls                                    15:13:29
acquirer_api                    financials_common               platform_api
acquirer_client                 financials_server               present
acquirer_common                 i2c_server                      present_client
acquirer_server                 integration                     promocard_client
admin_api                       integration_client              promocard_common
api                             integration_library             promocard_server
assets                          integrations_web                promocard_web
assets_client                   iso8583                         receipts
collect                         iso_auth_server                 receipts_client
collect_client                  kyc_client                      report_client
common                          kyc_common                      report_common
comms                           kyc_server                      report_server
comms_client                    ledger_v2_client                rewards_client
comms_web                       ledger_v2_common                rewards_common
customer                        ledger_v2_server                rewards_server
customer_client                 marqeta_server                  search_client
customer_common                 mastercard_clearing_processor   search_common
e2e_server                      messaging                       search_server
export_client                   micro                           underwriting_client
export_common                   network_client                  underwriting_common
export_server                   network_common                  underwriting_server
financials_client               network_server
~/s/g/v/c/apps on feature/promocard ⇄                                       15:13:30
```

Brex 0.1

Brex 1.0

- Umbrella
- “Service Oriented Architecture”
- About ~10 services

BREX

Distributed but no so much

- Our RPC implementation was...
- Kernel.apply

Brex 1.0

Brex 1.0

- Umbrella
- Service Oriented Architecture
- About ~20 services

Brex 1.0

- Cross service communication using RabbitMQ
  - Ensures retries and delivery
  - Will become a bottleneck

What about infrastructure?

Cluster
Namespaces
Nodes
Persistent Volumes
Roles
Storage Classes
Namespace
e-core-production
Overview
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Discovery and Load Balancing
Ingresses
Services
Config and Storage
Config Maps

core-api	app: core-api chart: core-api-1.0.0 heritage: Tiller release: core-production-core-api	4 / 4	18 hours	088757392028.dkr.ecr.us-west-2.amazonav	⋮
core-db-proxy	app: core-db-proxy chart: core-db-proxy-1.0.0 heritage: Tiller release: core-production-core-db-proxy	2 / 2	a day	quay.io/brex/pgbouncer:1.7.2	⋮
core-promocard-web	app: core-promocard-web chart: core-promocard-web-1.0.0 heritage: Tiller release: core-production-core-promocard...	4 / 4	a day	088757392028.dkr.ecr.us-west-2.amazonav	⋮
core-promocard	app: core-promocard chart: core-promocard-1.0.0 heritage: Tiller release: core-production-core-promocard	4 / 4	a day	088757392028.dkr.ecr.us-west-2.amazonav	⋮
core-search-server	app: core-search-server chart: core-search-1.0.0 heritage: Tiller release: core-production-core-search	4 / 4	a month	088757392028.dkr.ecr.us-west-2.amazonav	⋮
core-search-indexer	app: core-search-indexer chart: core-search-1.0.0 heritage: Tiller release: core-production-core-search	6 / 6	a month	088757392028.dkr.ecr.us-west-2.amazonav	⋮
core-i2c	app: i2c chart: i2c-1.0.0 heritage: Tiller release: core-production-core-i2c	4 / 4	2 months	088757392028.dkr.ecr.us-west-2.amazonav	⋮
core-iso-auth	app: core-iso-auth chart: core-iso-auth-1.0.0 heritage: Tiller	6 / 6	2 months	088757392028.dkr.ecr.us-west-2.amazonav	⋮

Infrastructure

- Kubernetes
- Containers
- AWS
- We still do mix

What’s next?

Brex Architecture 1.1

Guaranteeing Consistency

- Eventual consistency
- Data propagation events
- Doesn’t require ACID compliant databases

BREX

Execution Model

- Change local data
- Propagate side effects

BREX

Idempotence

- Most important aspects
- Requisite for eventual consistency

Build System

BREX

Build System

- Integration with Bazel?
- Build releases?
- Smart build plans

Global Query System

BREX

Global Query System

- GraphQL but for backend services
- Ecto Integration?
- Auto generated APIs?

Authentication
Authorization
Accounting

The First Corporate Card for Startups

COME JOIN US

We are hiring!

https://brex.com/careers