How we use  at 

Joel Jacobson

PgConf 2014, October 24, Madrid

Trustly

Trustly

What is Trustly?

- Stockholm-based payments company, founded in 2008
- Simplified business model: Moves money from bank account A to B
- 8 000 000 physical real-time bank transfers per year (no batches)
- 65 employees across Stockholm, Barcelona and Malta

- Integrated with 52 banks in 7 countries
- Relies on PostgreSQL for all data and business logics
- PostgreSQL contributor: 21 committed patches developed by our engineers

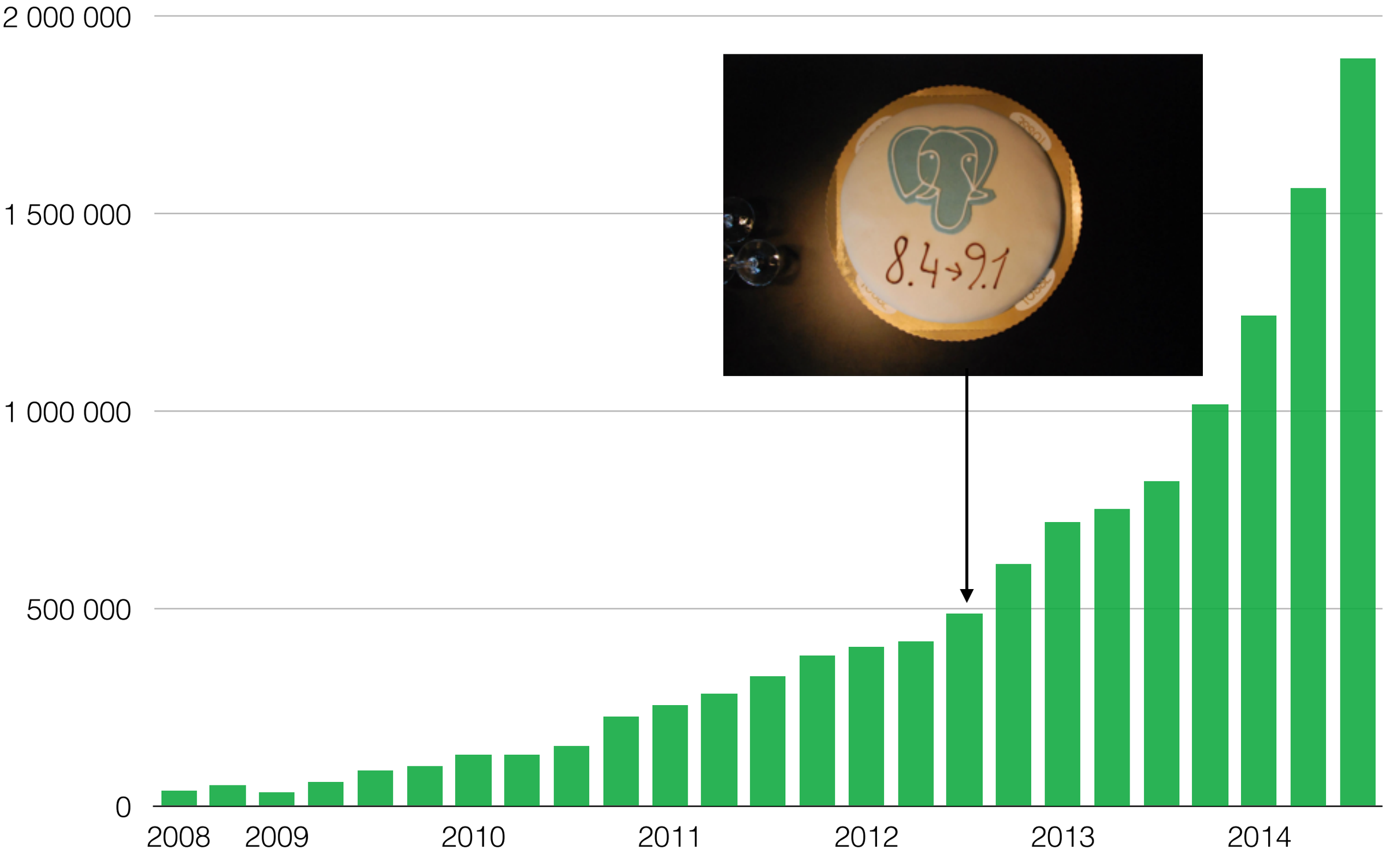
PostgreSQL contributions

(extracted from major release notes)

- **Marko Tiikkaja (johto):**
  - Split the processing of INSERT/UPDATE/DELETE operations out of execMain.c
  - Allow data-modification commands (INSERT/UPDATE/DELETE) in WITH clauses
  - Allow WITH clauses to be attached to INSERT, UPDATE, DELETE statements
  - Fix ordinary queries with rules to use the same snapshot behavior as EXPLAIN ANALYZE
  - Add transaction-level advisory locks
  - Make EXPLAIN ANALYZE report the number of rows rejected by filter steps
  - Cause pg_get_viewdef() to start a new line by default after each SELECT target list entry and FROM entry
  - Add cardinality() function for arrays
  - Add option print_strict_params to output parameters passed to queries generating STRICT errors
  - Add variables plpgsql.extra_warnings and plpgsql.extra_errors to enable additional PL/pgSQL warnings and errors
  - Make psql \do+ display the functions which implement the operators
- **Steeve Lennmark:**
  - Allow pg_basebackup to relocate tablespaces in the backup copy
- **Joel Jacobson:**
  - Add pg_stat_xact_* statistics functions and views
  - Make pg_dump output functions in a more predictable order

Trustly

Trustly bank transfers quarterly (linear)



Trustly

Example merchants

betsson

blocket

SAMSUNG

tasspass

facebook

GROUPON

BARNSKO SPECIALISTEN

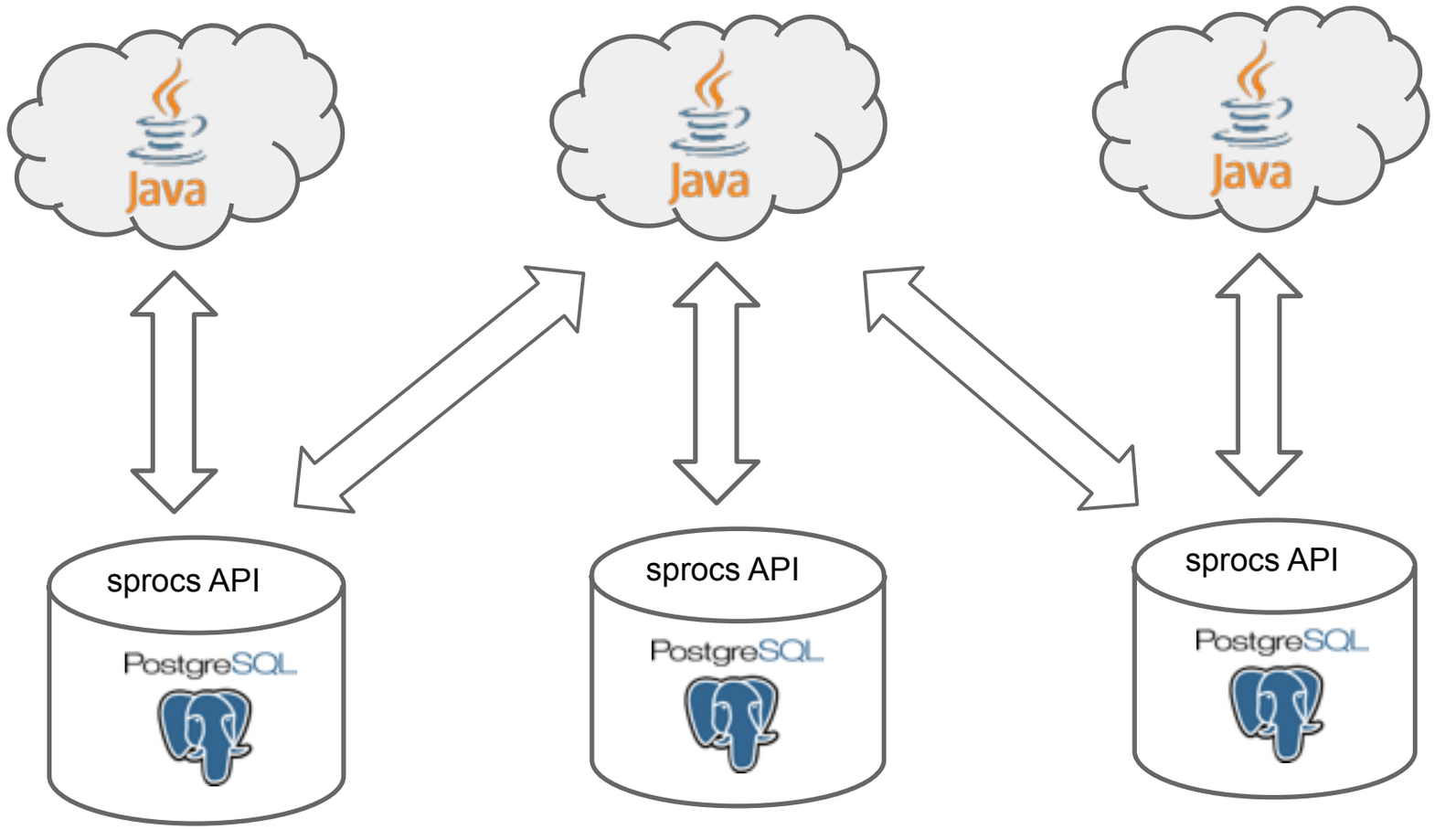
UNIBET

Tripwell

PayPal

How Trustly and Zalando are using PostgreSQL

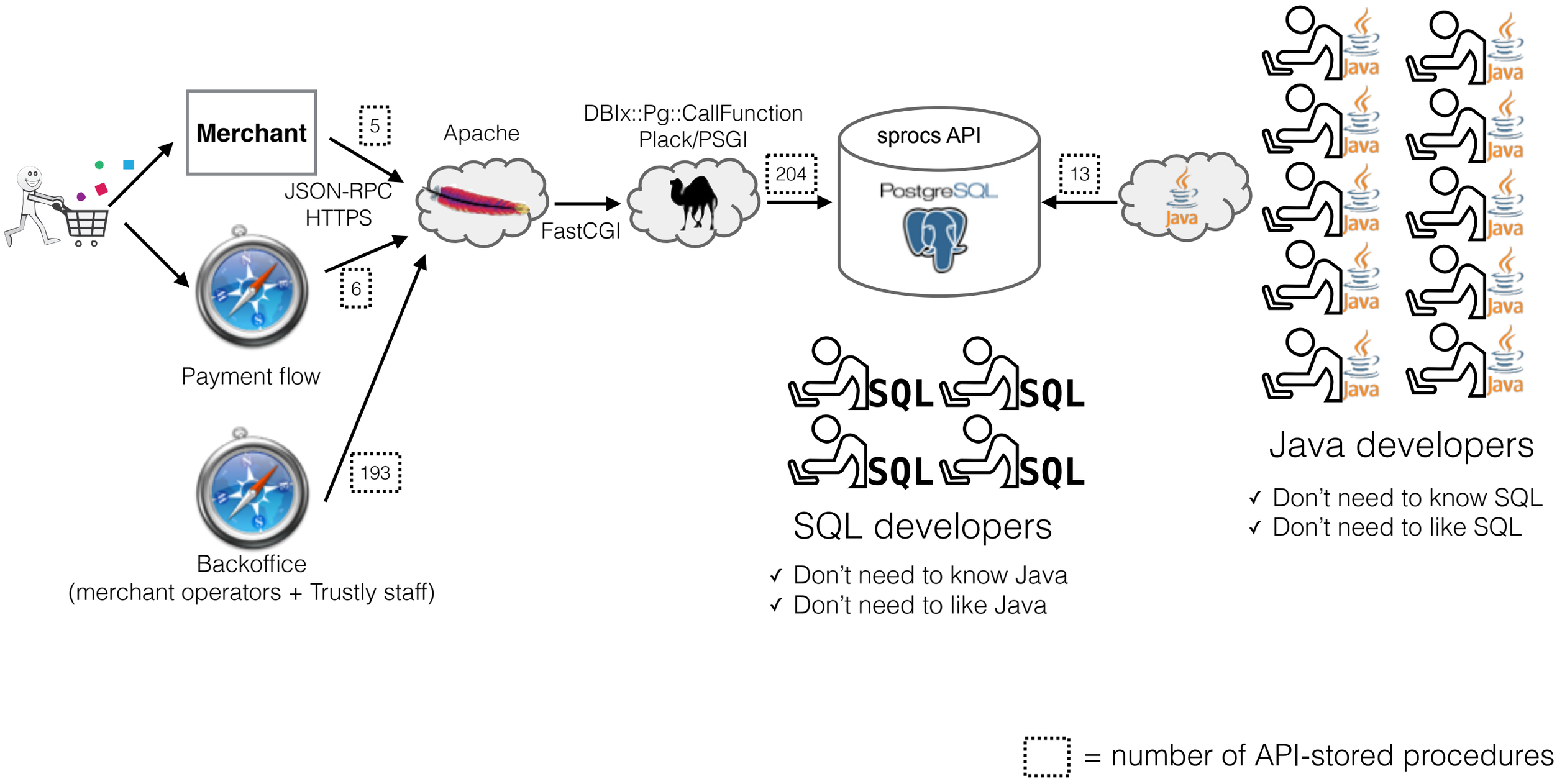
Applications access data in PostgreSQL databases via calls to stored procedures.



(slide copied from the excellent 2014.pgconf.eu presentation by Alexey Klyukin @ Zanaldo)

Trustly

Stored Procedure API



Trustly

Our database…

- 560 GB data (317 GB tables + 243 GB indexes)
- 372 tables
- Efficient data footprint (1 INSERT = €0.01 revenue)

…is not just data

- 1 530 functions (80% plpgsql, 7% plperl/u, 13% sql)
- 90 715 lines / 3.7MB of code (~60 lines / function)
- ~9 production database deployments per day

Trustly

Avoiding Game Over

- Synchronous replication between two datacenters
- Async replica to take over sync-role
- Standard Point-In-Time-Recovery setup for backups
- Automatic backup testing

PostgreSQL tools we use

https://github.com/trustly/fdiff	Tool to reduce risk for human errors when deploying functions and views in a PostgreSQL database.
https://github.com/trustly/dbix-pg-callfunction	Simple interface for calling PostgreSQL functions from Perl
https://github.com/johto/pgcov	PL/pgSQL test coverage analysis
https://github.com/johto/call_graph	A PostgreSQL extension for automatically creating function call graphs
https://github.com/okbob/plpgsql_check	Static code analyzer to find errors in PL/pgSQL source code
schema_diff (Marko Tiikkaja, johto)	Checks if git repo of database schema is identical with production database
plpgsql_column_usage (Marko Tiikkaja, johto)	Static code analyzer showing what functions INSERT/UPDATE/DELETE what table columns

JSON API

```
{
    "method": “sell_currency”,
    "params": {
        "username": "joel",
        "password": "07cc2ef0-6c24-46fb-9323-ae2c6e586f64",
        "sellamount": 10,
        "sellcurrency": "SEK",
        "buycurrency": "EUR"
   },
    "version": 1.1
}
```

https

Apache

mod_fastcgi

local domain socket

plackup -s FCGI

PSGI/Plack

DBIx::Pg::CallFunction

pg_catalog.pg_proc

postgresql/5432 hostssl cert

```
SELECT * FROM public.sell_currency(
    username     := ‘joel’,
    password     := ‘07cc2ef0-6c24-46fb-9323-ae2c6e586f64’,
    sellamount   := 10,
    sellcurrency := ‘SEK’,
    buycurrency  := ‘EUR’
);
```

Trustly

JSON API

- JSON request method + params —> pg_proc.proname + pg_proc.proargnames
- JSON response data structure <— pg_proc.prorettype + pg_proc.proretset
  - `RETURNS text -> “foo”`
  - `RETURNS text[] -> [“foo”,”bar”,”baz”]`
  - `RETURNS RECORD -> {“firstname”:”Joel”,”lastname”:”Jacobson”}`
  - `RETURNS SETOF RECORD -> [`

```
    {“userid”:1,”username”:”alice”},
    {“userid”:2,”username”:”bob”}
]
```

  - `RETURNS json -> {“foo”:{“bar”:”baz”}}`

JSON API

- Fast and easy deployment - no downtime:
    1. `CREATE OR REPLACE FUNCTION …`
        - `SET search_path TO public, pg_temp`
        - `SECURITY DEFINER`
    2. `GRANT EXECUTE ON FUNCTION … TO api;`
- Offloading read-only queries to the slave (`STABLE`)
- API-errors logged to database

Trustly

Database schema version control

(of all objects and base data)

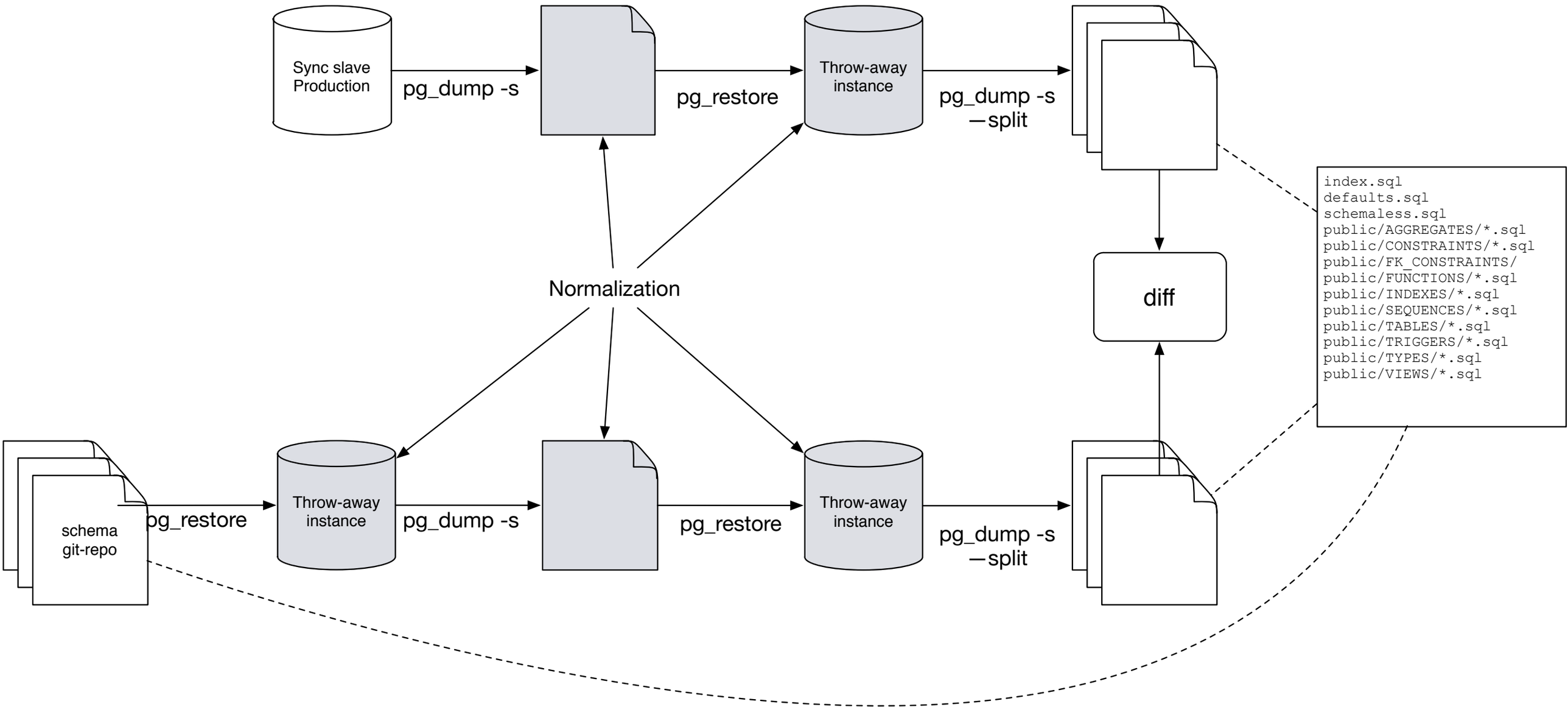
`./[namespace]/[object type]/[object name].sql`

```
./install-osx.sh
./install-ubuntu.sh
./install.sh
./index.sql
./base_data
./base_data/000.public.foo.csv
./base_data/001.public.bar.csv
./base_data/002.public.baz.csv
./base_data/restore.sql
./defaults.sql
./schemaless.sql
./EXTENSIONS/foo.sql
./public/operators.sql
./public/AGGREGATES/foo.sql
./public/CONSTRAINTS/foo_pkey.sql
./public/FK_CONSTRAINTS/foo_barid_fkey.sql
./public/FUNCTIONS/foo.sql
./public/INDEXES/index_foo_barid.sql
./public/SEQUENCES/seqfoo.sql
./public/TABLES/foo.sql
./public/TABLES/bar.sql
./public/TRIGGERS/foo.sql
./public/TYPES/foo.sql
./public/VIEWS/foo.sql
./t/00-foo.t
./t/01-bar.t
./test/continuous_delivery/1.sql
./test/continuous_delivery/2.sql
./test/continuous_delivery/3.sql
./test/index.sql
```

```
SET client_encoding TO 'UTF8';

\i schemaless.sql
\i operators.sql
\i EXTENSIONS/foo.sql
\i public/TYPES/foo.sql
\i public/FUNCTIONS/foo.sql
\i public/AGGREGATES/foo.sql
\i public/SEQUENCES/foo.sql
\i public/TABLES/foo.sql
\i public/TABLES/bar.sql
\i public/VIEWS/bar.sql
\i defaults.sql
\i public/CONSTRAINTS/foo_pkey.sql
\i public/FK_CONSTRAINTS/foo_barid_fkey.sql
\i public/INDEXES/index_foo_barid.sql
```

schema_diff


Sync slave
Production
pg_dump -s
pg_restore
Throw-away
instance
pg_dump -s
—split
index.sql
defaults.sql
schemaless.sql
public/AGGREGATES/*.sql
public/CONSTRAINTS/*.sql
public/FK_CONSTRAINTS/
public/FUNCTIONS/*.sql
public/INDEXES/*.sql
public/SEQUENCES/*.sql
public/TABLES/*.sql
public/TRIGGERS/*.sql
public/TYPES/*.sql
public/VIEWS/*.sql
diff
Normalization
schema
git-repo
pg_restore
Throw-away
instance
pg_dump -s
pg_restore
Throw-away
instance
pg_dump -s
—split

Trustly

fstage() -> deploy -> fdiff()

```
Joels-MacBook-Pro:schema joel$ git show --show-signature 45b13a8428441632008b81cde55416603ec7aa61
commit 45b13a8428441632008b81cde55416603ec7aa61
gpg: Signature made Tue Oct 21 12:58:14 2014 CEST using RSA key ID 651F4D05
gpg: Good signature from "SE198210220136" [ultimate]
Author: Joel Jacobson <joel@trustly.com>
Date:   Tue Oct 21 12:58:07 2014 +0200

    Validate IBAN length (not only the check-digits)

diff --git a/public/FUNCTIONS/validate_iban.sql b/public/FUNCTIONS/validate_iban.sql
index 6de4a0a..7336b2b 100644
--- a/public/FUNCTIONS/validate_iban.sql
+++ b/public/FUNCTIONS/validate_iban.sql
@@ -1,7 +1,7 @@
 SET search_path TO 'public', pg_catalog;

 CREATE OR REPLACE FUNCTION validate_iban(_iban character varying) RETURNS boolean
-    LANGUAGE plpgsql IMMUTABLE
+    LANGUAGE plpgsql STABLE
     AS $_$
 DECLARE
 checkdigit  integer := 0;
@@ -14,6 +14,9 @@ BEGIN
     RETURN FALSE;
 END IF;
+IF length(_IBAN) <> (SELECT IBANTotalLength FROM IBANStructure WHERE IBANCountryCode = substr(_IBAN,1,2)) THEN
+    RETURN FALSE;
+END IF;
 str := substr(_IBAN,5,length(_IBAN)-4) || substr(_IBAN,1,4);
 FOR ch IN SELECT * FROM regexp_split_to_table(str,'') LOOP
     IF ch BETWEEN 'A' AND 'Z' THEN
```

fstage() -> deploy -> fdiff()

```
pg1:trustly=> BEGIN;
BEGIN
pg1:trustly=> SELECT fstage();
 fstage
--------
 t
(1 row)

pg1:trustly=>* \i /Users/trustly/src/schema/public/FUNCTIONS/validate_iban.sql
SET
CREATE FUNCTION
ALTER FUNCTION
pg1:trustly=>* SELECT fdiff();
                                                        fdiff
---------------------------------------------------------------------------------------------------------------------
                                                                                                                    +
                                                                                                                    +
 +------------------------------+                                                                                   +
 | Updated or replaced functions |                                                                                  +
 +------------------------------+                                                                                   +
                                                                                                                    +
 Schema.....................: public                                                                                +
 Name.......................: validate_iban                                                                         +
 Argument data types........: _iban character varying                                                               +
 Result data type...........: boolean                                                                               +
 Language...................: plpgsql                                                                               +
 Type.......................: normal                                                                                +
 Volatility.................- IMMUTABLE                                                                             +
 Volatility.................+ STABLE                                                                                +
 Security definer...........: false                                                                                 +
 Configuration parameters...:                                                                                       +
 Owner......................: gluepay                                                                               +
 13 +                                                                                                               +
 13 + IF length(_IBAN) <> (SELECT IBANTotalLength FROM IBANStructure WHERE IBANCountryCode = substr(_IBAN,1,2)) THEN+
                                                                                                                    +
 14 +                                                                                                               +
 14 +     RETURN FALSE;                                                                                             +
                                                                                                                    +
 15 +                                                                                                               +
 15 + END IF;                                                                                                       +
(1 row)

pg1:trustly=>* COMMIT;
COMMIT
```

Testing of Stored Procedures

https://github.com/trustly/dbix-pg-callfunction

```perl
#!/usr/bin/perl
use strict;
use warnings;

use DBI;
use DBIx::Pg::CallFunction;
use JSON qw(to_json);

use Test::More;
use Test::Deep;
use Test::Exception;

my $dbh = DBI->connect("dbi:Pg:", '', '', {pg_enable_utf8 => 1,
AutoCommit => 1});
my $pg = DBIx::Pg::CallFunction->new($dbh);
```

1

```perl
throws_ok {
    $pg->verify_user_auth_session({
        _username => $username,
        _sessionuuid => '00000000-0000-0000-0000-000000000000',
        _host => $host,
        _function => 'Change_User_Password'
    })
} qr/ERROR_INVALID_SESSIONUUID/, 'ERROR_INVALID_SESSIONUUID';
```

2

```perl
ok(
    $pg->verify_notification_response({
        _notificationid => $notificationid,
        _jsondata => to_json($response),
        _uuid => $uuid,
        _signature => $signature
    }),
    "Verify_Notification_Response"
);
```

3

```perl
is(
    $pg->get_payment_amount({_orderid => $orderid}),
    123.45,
    "Get_Payment_Amount"
);
```

4

```perl
cmp_deeply(
    $pg->get_order_chain_balances({
        _orderid => $deposit_orderid
    }),
    [
        {
            currency => $paymentcurrency,
            amount => -$paymentamount,
            accountnumber => 991106,
            accountname => 'DEPOSIT_CONFIRMED'
        },
        {
            currency => $apicurrency,
            amount => $apiamount+0,
            accountnumber => 992102,
            accountname => 'CREDIT_SENDING'
        }
    ],
    "Get_Order_Chain_Balances"
);
```

Trustly

https://github.com/johto/call_graph

add_organization

5 calls	164.56 ms total	32.91 ms average	First call 2014-09-23 15:23:07	Last call 2014-09-23 15:23:11

No calls today
No calls in the current hour
No calls in the previous hour

read-write tables	seq_scan	seq_tup_read	idx_scan	idx_tup_read	n_tup_ins	n_tup_upd	n_tup_del
organizations	0	0	5	0.00	5	0	0
useraccesslog	5	501.80	5	0.00	5	0	0
read-only tables	**seq_scan**	**seq_tup_read**	**idx_scan**	**idx_tup_read**	**n_tup_ins**	**n_tup_upd**	**n_tup_del**
users	0	0	55	0.91	0	0	0
functions	0	0	20	1.00	0	0	0
loginattempts	0	0	15	0.00	0	0	0
userfunctions	0	0	5	1.00	0	0	0
userhosts	5	128.00	0	0	0	0	0

View table information

get_session_cookie

4 calls	134.50 ms total	33.62 ms average	First call 2014-09-23 15:23:06	Last call 2014-09-23 15:23:07

No calls today
No calls in the current hour
No calls in the previous hour

read-write tables	seq_scan	seq_tup_read	idx_scan	idx_tup_read	n_tup_ins	n_tup_upd	n_tup_del
useraccesslog	4	488.00	4	0.50	4	0	0
userauthsessions	0	0	0	0	4	0	0
read-only tables	**seq_scan**	**seq_tup_read**	**idx_scan**	**idx_tup_read**	**n_tup_ins**	**n_tup_upd**	**n_tup_del**
users	0	0	48	0.92	0	0	0
functions	0	0	16	1.00	0	0	0
loginattempts	0	0	12	0.00	0	0	0
userfunctions	0	0	4	0.00	0	0	0
userhosts	4	123.50	0	0	0	0	0

View table information

new_client_response

85 calls	95.70 ms total	1.13 ms average	First call 2014-09-23 15:22:58	Last call 2014-09-23 15:23:56

No calls today
No calls in the current hour
No calls in the previous hour

read-write tables	seq_scan	seq_tup_read	idx_scan	idx_tup_read	n_tup_ins	n_tup_upd	n_tup_del
orders	0	0	340	1.00	0	85	0
ordersteps	0	0	255	1.00	0	85	0
read-only tables	**seq_scan**	**seq_tup_read**	**idx_scan**	**idx_tup_read**	**n_tup_ins**	**n_tup_upd**	**n_tup_del**
orderstatuses	0	0	174	1.00	0	0	0
ordersteptypes	0	0	93	1.00	0	0	0
sessions	85	265.58	0	0	0	0	0

View table information

Trustly

https://github.com/okbob/plpgsql_check

lineno	statement	sqlstate	message	detail	hint	level	position	
29	SQL statement	42702	column reference "currency" is ambiguous			error	968	WITH CurrencyBalances AS (SELECT TransferStateBalances.Currency, SUM(TransferStateBalances.B INNER JOIN TransferBankAccounts ON (TransferBankAccounts.Transfe INNER JOIN BankNumbers ON (BankNumbers.BankNumber = TransferI AND BankNumbers.ClearingHouseID = TransferBankAccounts.ClearingH INNER JOIN TransferStateBalances ON (TransferStateBalances.ClearingI AND TransferStateBalances.BankID = BankNumbers.BankID AND TransferStateBalances.PersonID IS NULL AND TransferStateBalances.UserID IS NULL) WHERE SettlementAccounts.TransferBankAccountID = _TransferBankA GROUP BY Currency) SELECT SUM(Balance*FX(Currency,_InTheAirLimitCurrency)) FROM
		0A000	could not determine actual argument type for polymorphic function "array_empty"			error	0	
35	RETURN	00000	unused declared variable	variable _clientresponse declared on line 7		warning	0	
103	SQL statement	42P01	relation "pg_temp.tmptransfers" does not exist			error	13	INSERT INTO pg_temp.tmpTransfers SELECT TransferID, FromBank, Datestamp, Settled, COALESCE(NOT Settled AND -- rule #1 (Datestamp < BoundaryMin OR -- rule #2 Datestamp < BoundaryMax), FALSE) AS Failed, -- rule #3 is checked below COALESCE(Datestamp < BoundaryMax, FALSE) AS IsWithinHorizon FROM (SELECT TransferID, FromBank, Datestamp, Settled, min(CASE WHEN Settled THEN Datestamp END) OVER w - interval '24 max(SecondToLast) OVER w - interval '24 hours' AS BoundaryMax FROM Auto_Fail_Missing_Payments_Show_Horizon(_StatementID, _Ho WINDOW w AS (PARTITION BY FromBank)) ss
36	RETURN	00000	unused declared variable	variable _timepassed declared on line 6		warning	0	
36	RETURN	00000	unused declared variable	variable _adid declared on line 4		warning	0	
160	FOR over SELECT rows	42P01	relation "pg_temp.autosettlecalculations" does not exist			error	84	SELECT AutoSettleCalculations.Currency, AutoSettleCalculations.Amou FROM pg_temp.AutoSettleCalculations WHERE AutoSettleCalculations.Amount < 0.00
31	RETURN	00000	unused declared variable	variable _clientresponse		warning	0	

Trustly

https://github.com/johto/pgcov

function await_order_approval_if_required(bigint, integer):

```
 1
 2  DECLARE
 3  _ApprovalRequired boolean;
 4  BEGIN
 5  PERFORM Check_Order_Worker(_OrderID, _WorkerID);
 6
 7  _ApprovalRequired := Is_Withdrawal_Approval_Required(_OrderID,_WorkerID);
 8
 9  IF _ApprovalRequired IS TRUE THEN
10      IF Get_Order_Access_Status(_OrderID,_WorkerID) = 'DENIED' THEN
11          -- Order permanently DENIED, no need to wait, fail it at once.
12          RETURN -1;
13      END IF;
14      -- We need to await approval
15      PERFORM Wait_Event(_OrderID,_WorkerID);
16      RETURN 1;
17  ELSIF _ApprovalRequired IS FALSE THEN
18      -- Approval not required, approve withdrawal
19      PERFORM _Approve_Order(_OrderID);
20      RETURN 0;
21  ELSE
22      RAISE EXCEPTION 'ERROR_WTF _ApprovalRequired %', _ApprovalRequired;
23  END IF;
24
25  END;
26
```

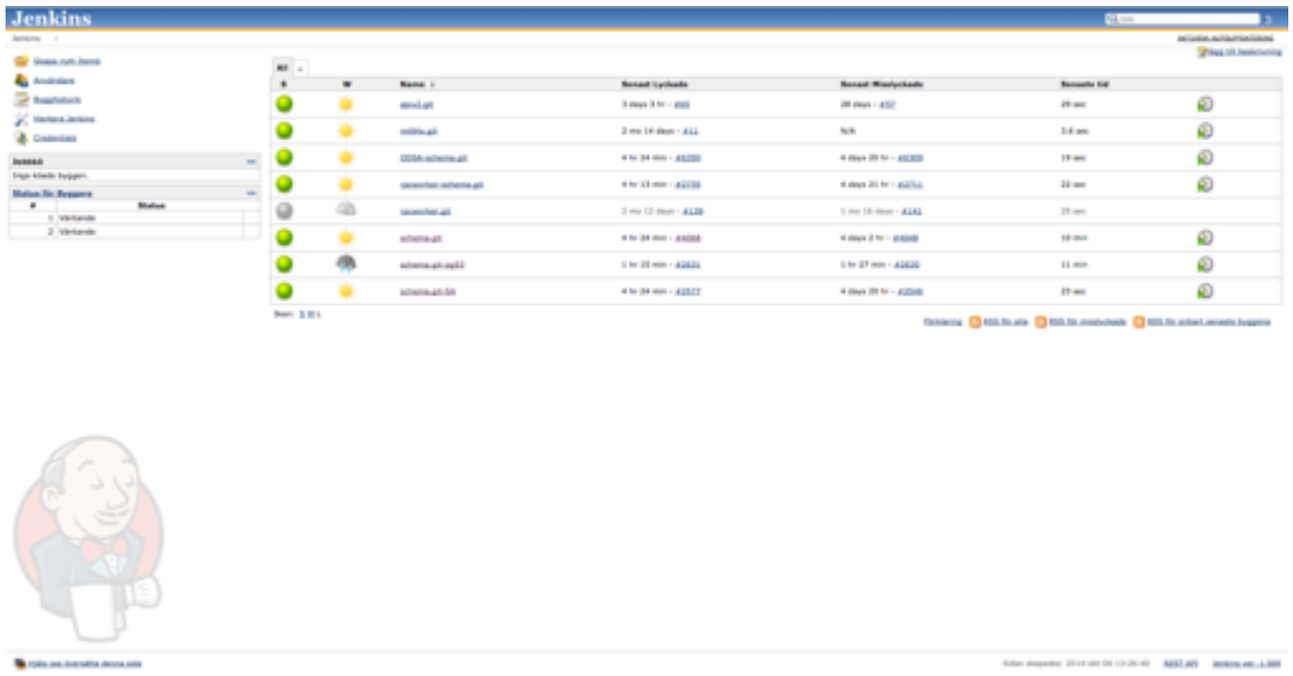
Trustly

plpgsql_column_usage

fn	operation	tablename	column
autosettle.calculate_automatic_settlements	UPDATE	autosettle.configuration	configuration.lastsettlementattempt
autosettle.process_automatic_settlements	UPDATE	autosettle.settlements	settlements.bankwithdrawalid
fees.bill_merchant_monthly_fee_calculation	UPDATE	fees.merchantcalculatedfees	merchantcalculatedfees.eventid
fees.archive_merchant_monthly_fee_calculation	UPDATE	fees.merchantmonthlyfeecalculations	merchantmonthlyfeecalculations.archived
fees.bill_merchant_monthly_fee_calculation	UPDATE	fees.merchantmonthlyfeecalculations	merchantmonthlyfeecalculations.billed
fees.calculate_merchant_monthly_fees	UPDATE	fees.merchantmonthlyfeecalculations	merchantmonthlyfeecalculations.succeeded
fees.invalidate_merchant_monthly_fee_calculations	UPDATE	fees.merchantmonthlyfeecalculations	merchantmonthlyfeecalculations.invalidated
fees.invalidate_merchant_monthly_fee_calculations	UPDATE	fees.merchantmonthlyfeecalculations	merchantmonthlyfeecalculations.invalidated
fees.archive_merchant_monthly_fee_version	UPDATE	fees.merchantmonthlyfees	merchantmonthlyfees.modificationdate
fees.remove_merchant_monthly_fee	UPDATE	fees.merchantmonthlyfees	merchantmonthlyfees.effectiveto
fees.update_merchant_monthly_fee	UPDATE	fees.merchantmonthlyfees	merchantmonthlyfees.amount
fees.update_merchant_monthly_fee	UPDATE	fees.merchantmonthlyfees	merchantmonthlyfees.effectiveto
fees.update_merchant_monthly_fee	UPDATE	fees.merchantmonthlyfees	merchantmonthlyfees.additionalinformation
fees.update_merchant_monthly_fee	UPDATE	fees.merchantmonthlyfees	merchantmonthlyfees.chargeaccountuserid
fees.update_merchant_monthly_fee	UPDATE	fees.merchantmonthlyfees	merchantmonthlyfees.currency
fees.update_merchant_monthly_fee	UPDATE	fees.merchantmonthlyfees	merchantmonthlyfees.modificationdate
aml.set_address	INSERT	aml.addresses	
aml.set_birth	INSERT	aml.births	
aml.set_citizen	INSERT	aml.citizens	
aml.set_entity	INSERT	aml.entities	
aml.get_linkid	INSERT	aml.links	
aml.set_name	INSERT	aml.names	
aml.set_passport	INSERT	aml.passports	
aml.set_entity	INSERT	aml.remarks	
fees.remove_merchant_commitment_fee_group_member	DELETE	fees.merchantcommitmentfeegroupmembers	
fees.remove_merchant_monthly_fee	DELETE	fees.merchantcommitmentfeegroupmembers	
fees.remove_merchant_monthly_fee	DELETE	fees.merchantmonthlyfees	

Trustly

Jenkins (Continuous Integration)



Automatically runs all of it
for each new pushed commit:

- schema_diff
- test suite
- call_graph
- plpgsql_check
- pgcov
- plpgsql_column_usage

We're hiring!

Work 100% with  at 

- ☑ No NoSQL
- ☑ No Windows
- ☑ No deadlines
- ☑ Almost no meetings
- ☑ No SCRUM
- ☑ No 9 - 5 working hours

Email me:

Joel Jacobson <joel at trustly.com>

PgConf 2014, October 24, Madrid