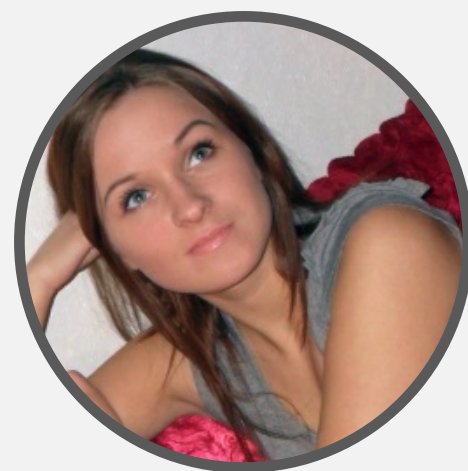




Камынина Дарья
IQ Option

РАЗРАБОТКА И НЕПРЕРЫВНАЯ ИНТЕГРАЦИЯ ХРАНИМЫХ ПРОЦЕДУР

<https://github.com/kamynina/pgday-example>

On-line брокер IQ Option



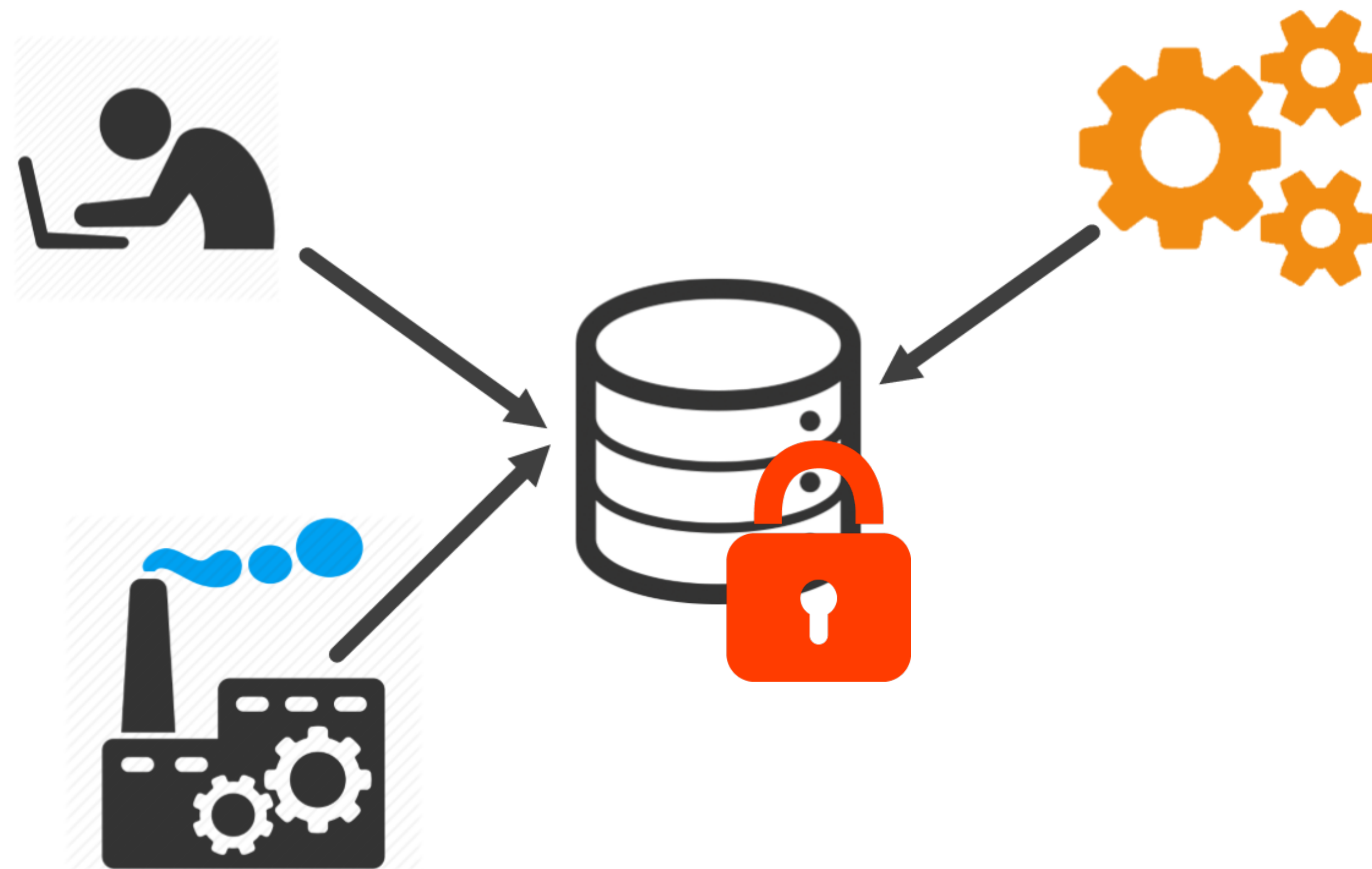
TRADE IN THE MOMENT
INVEST IN THE FUTURE

Cutting edge solutions for stocks (CFDs), ETFs, FX
and Options

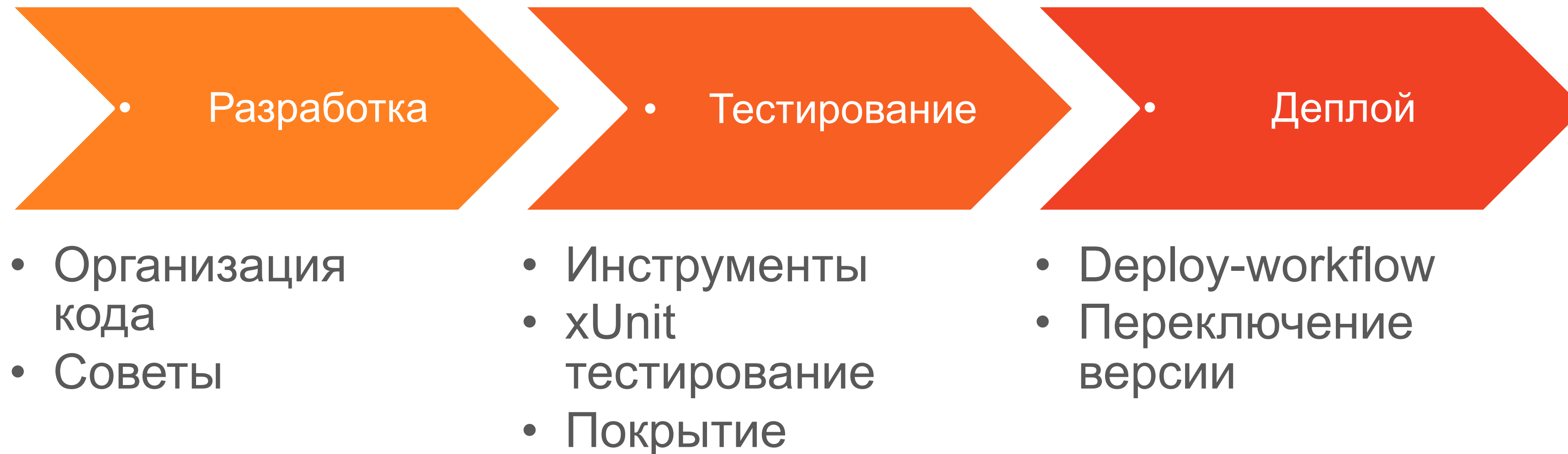
19,919,668 accounts across the world

- 2500 сделок в минуту
- 40К запросов в секунду на платформу; в пике 150К
- Размер базы: 7,2 Тб

Архитектура



Цикл разработки



РАЗРАБОТКА

Организация кода, советы по написанию хранимых процедур



Разработка: организация кода

The screenshot shows a code editor with a project structure on the left and SQL code on the right.

Project Structure (Left Panel):

- pgday-example ~/Documents/pg
 - src
 - api
 - functions
 - iterate.sql
 - register_user.sql
 - types
 - views
 - _init.sql
 - init.sql
 - tests
 - api
 - functions
 - test_iterate_array_em
 - test_register_user.sql
 - types
 - views
 - _init.sql
 - init.sql
 - init_db.sql
 - .gitignore
 - .travis.yml
 - deploy.sh
 - docker-compose.yml

SQL Code (Right Panel):

```
BEGIN;  
CREATE SCHEMA api AUTHORIZATION example;  
GRANT ALL ON SCHEMA api TO example;  
SET search_path = api, public;  
  
\ir types/user_role.sql  
  
\ir views/users.sql  
  
\ir functions/iterate.sql  
\ir functions/register_user.sql  
COMMIT;
```


Разработка: советы по написанию

- Входные параметры и локальные переменные процедур именуются, начиная с символа нижнее подчеркивание
- Пользователь, исполняющий ХП, не имеет прав на объекты БД (для выполнения исп. Security definer модификатор)
- *SET search_path*, где в конце указываем *pg_temp*, либо напрямую указываем схему в запросах

Разработка: советы по написанию (продолжение)

- Использовать `strict`
- Указывать тип изменчивости функции
`[VOLATILE|STABLE|IMMUTABLE]`
- Указывать количество строк, если
возвращается множество

ТЕСТИРОВАНИЕ

Unit-тестирование хранимых процедур; покрытие кода тестами



Тестирование: выбор инструмента

Perl

```
use Test::More tests => 3;
use DBI;

my $dbh = DBI->connect('dbi:Pg:dbname=try', 'postgres', " ");

# Start a transaction.
$dbh->begin;
END { $dbh->rollback; $dbh->disconnect; }
my $domain_id = 1;
my $src_id = 2;

# Insert stuff.
ok $dbh->do(
    'SELECT insert_stuff( ?, ?, ?, ?)',
    undef, 'www.foo.com', [1, 2, 3], $domain_id, $src_id
), 'Inserting some stuff should return true';

# Grab the stuff records.
ok my $stuff = $dbh->selectall_arrayref(q{
    SELECT stuff_id
    FROM domain_stuff
    WHERE domain_id = ?
    AND src_id = ?
    ORDER BY stuff_id
}, undef, $domain_id, $src_id), 'Fetch the domain stuff';

# Make sure we have the right stuff.
is_deeply $stuff, [ 1, 2, 3 ], 'The rows should have the right stuff';
```

PgTap

```
-- Start a transaction.
BEGIN;
SELECT plan( 2 );
\set domain_id 1
\set src_id 1

-- Insert stuff.
SELECT ok(
    insert_stuff( 'www.foo.com', '{1,2,3}', :domain_id, :src_id ),
    'insert_stuff() should return true'
);

-- Check for domain stuff records.
SELECT is(
    ARRAY(
        SELECT stuff_id
        FROM domain_stuff
        WHERE domain_id = :domain_id
        AND src_id = :src_id
        ORDER BY stuff_id
    ),
    ARRAY[ 1, 2, 3 ],
    'The stuff should have been associated with the domain'
);

SELECT * FROM finish();
ROLLBACK;
```

Тестирование: достоинства pgTap

- Простота
- Никаких дополнительных действий по работе с данными
- Только SQL – одна технология на все
- Test Anything Protocol – легкая интеграция в большинство CI
- Привычный для разработчиков xUnit-СТИЛЬ

Тестирование: xUnit тест

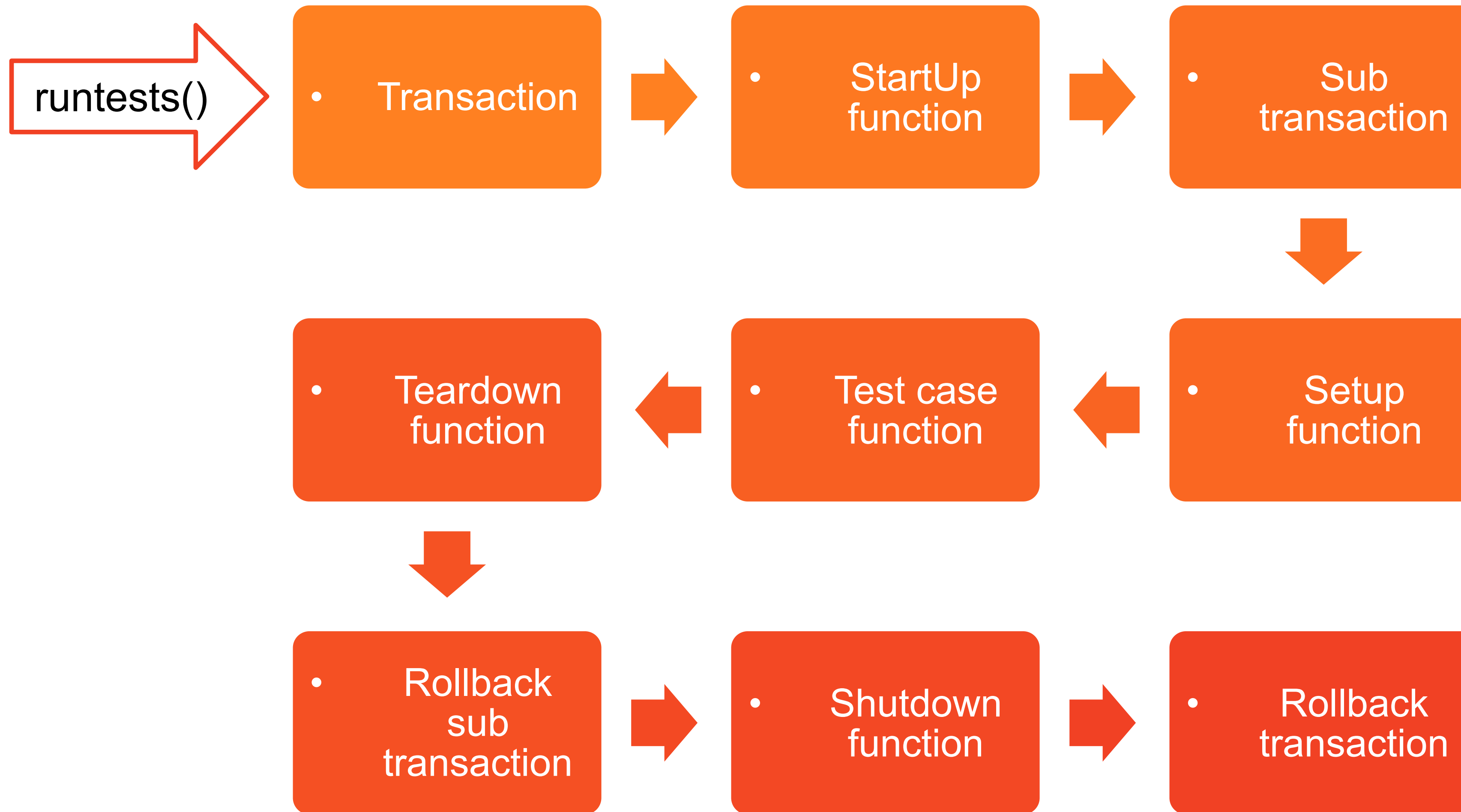
```
CREATE FUNCTION test_register_user() RETURNS  
SETOF TEXT LANGUAGE plpgsql AS $$  
DECLARE  
    _test_user_id BIGINT;  
BEGIN  
    _test_user_id := register_user('Wasya', '123');  
    RETURN NEXT cmp_ok(  
        _test_user_id,  
        '=',  
        curval('users_id_seq'),  
        'User created'  
    );  
END;  
$$;
```

docker image kamynina/pgtap

Тестирование: отчет

```
Running tests: /t/*.sql
runtests('test'::name); ..
    # Subtest: test.test_iterate_array_empty()
    ok 1
    1..1
ok 1 - test.test_iterate_array_empty
    # Subtest: test.test_register_user()
    ok 1 - Check row user is correct
    ok 2 - Return result equals users_id_seq
    1..2
ok 2 - test.test_register_user
1..2
ok
All tests successful.
Files=1, Tests=2,  0 wallclock secs ( 0.02 usr +  0.00 sys =  0.02 CPU)
Result: PASS
```

Тестирование: workflow



Тестирование: покрытие

- Инструмент – piggly
- Добавляет выражения RAISE WARNING в код процедур. После прогона тестов формируется вывод. На основании него генерируется отчет.

Ограничения

- Не вся грамматика PL/pgsql поддерживается
- Некорректная работа переменной FOUND
- Ветвления внутри SQL выражений (COALESCE, WHERE и др.) не отслеживаются

Тестирование: покрытие

PL/pgSQL Coverage Summary

Blocks	Loops	Branches	Block Coverage	Loop Coverage	Branch Coverage
10	1	3	80.00%	50.00%	66.67%

Procedure	Blocks	Loops	Branches	Block Coverage	Loop Coverage	Branch Coverage
iterate	6	1	2	100.00%	50.00%	75.00%
register_user	4	0	1	50.00%		50.00%

Generated by piggly 2.0.0 at July 07, 2017 00:44 UTC

docker image kamynina/pgpiggly

Тестирование: покрытие

api_1_1.iterate

Blocks	Loops	Branches	Block Coverage	Loop Coverage	Branch Coverage
6	1	2	100.00%	50.00%	75.00%

CREATE FUNCTION api_1_1.iterate (_items anyarray)
 RETURNS SETOF anyelement
 VOLATILE

1 DECLARE
 2 _array_min_index integer := ARRAY_LOWER(_items, 1);
 3 _array_max_index integer := ARRAY_UPPER(_items, 1);
 4 _loop_index integer;
 5 BEGIN
 6 IF _array_min_index is null
 7 THEN
 8 RETURN;
 9 END IF;
 10
 11 for _loop_index in array_min_index .. array_max_index LOOP
 12 IF _items [_loop_index] is not null
 13 THEN
 14 RETURN next _items [_loop_index];
 15 END IF;
 16 END LOOP;
 17 END;

Generated by piggly 2.0.0 at July 07, 2017 00:44 UTC

Index

1. loop always iterates more than once
 2. never evaluates false

ДЕПЛОЙ

Workflow, переключение версий



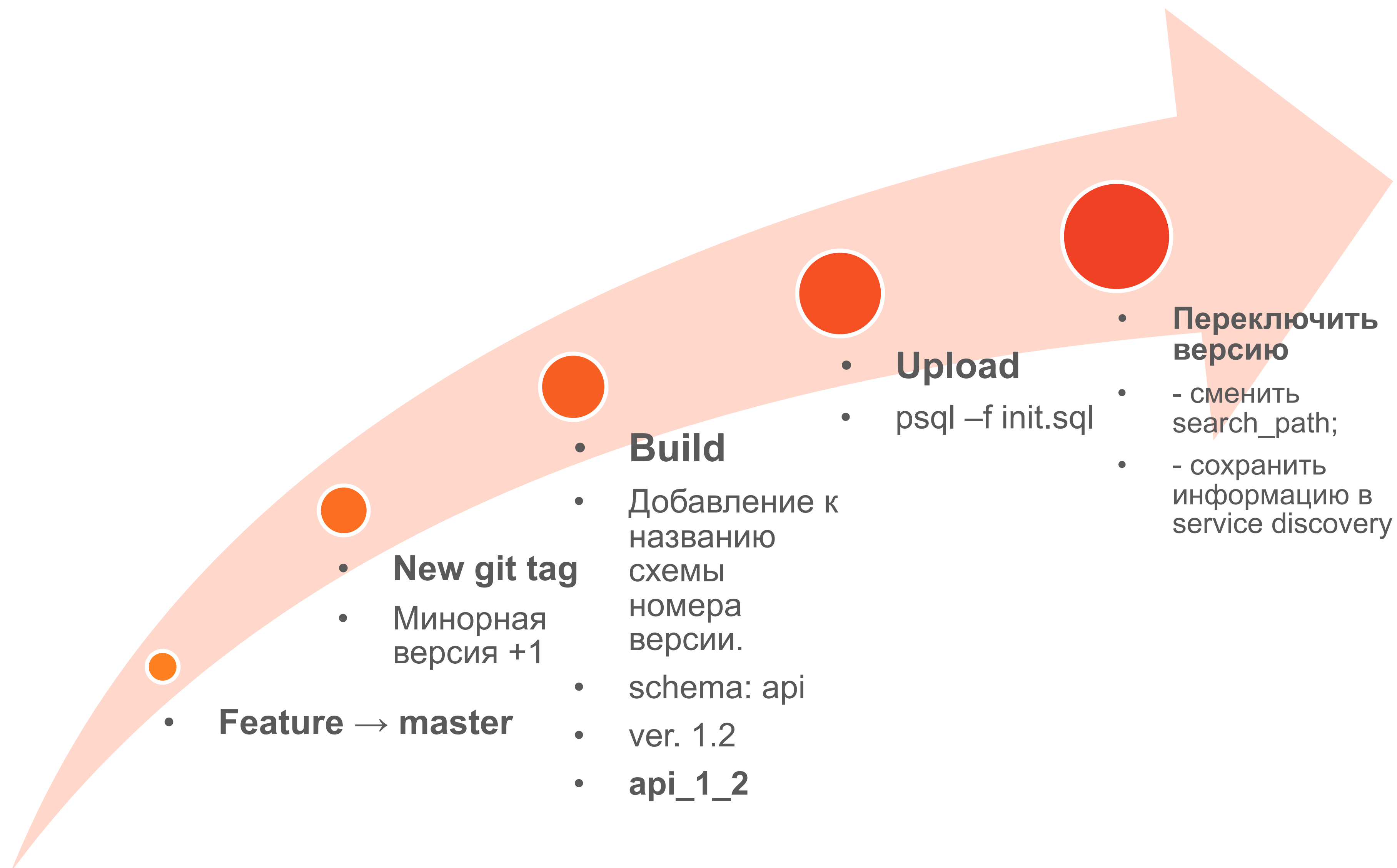
Деплой: требования

- **Версионирование** – разные пользователи ХП могут пользоваться различными версиями ХП (мажорными).
- **Прозрачность для клиента** – деплой не должен приводить к необходимости править клиентский код.
- **Rollback**
- **Отсутствие downtime**

Деплой: поиск решения

- новая ХП с именем, в котором присутствует номер версии.
 - сложности: непрозрачность для пользователя ХП либо необходимо писать прокси, который будет вызывать нужную версию.
- новая схема и переключение пользователя на эту схему

Деплой: workflow



Деплой: переключение версии

Варианты	Одновременно несколько мажорных версий	НЕ требуется сброс коннектов pgBouncer	Не требуется SUPERUSER	Оперативность перехода после деплоя	Кого затронем при деплое
В service discovery	✓	✓	✓	✗	Никого, пользователи сами
search_path для пользователя БД	✓	✗	✗	✓	Все пользователи
search_path для БД	✗	✗	✓	✓	1 ALTER DATABASE
search_path для кластера	✗	✓	✗	✓	Весь кластер

CI (Travis)

kamynina / pgday-example  build passing

Current Branches Build History Pull Requests More options 

✓ master Fix pipeline and change stream

 Commit faf1012
 Compare d24c0fa..faf1012
 Branch master

 dkamynina authored and committed

#7 passed

Ran for 42 sec
36 minutes ago

Restart build

🚩 This job ran on our Trusty, `sudo: required` environment which was updated on Wednesday, June 21st. Read all about it on our [blog](#) and take note that you can add `group: deprecated-2017Q2` in your `.travis.yml` file to use the previous version.

Спасибо за внимание!