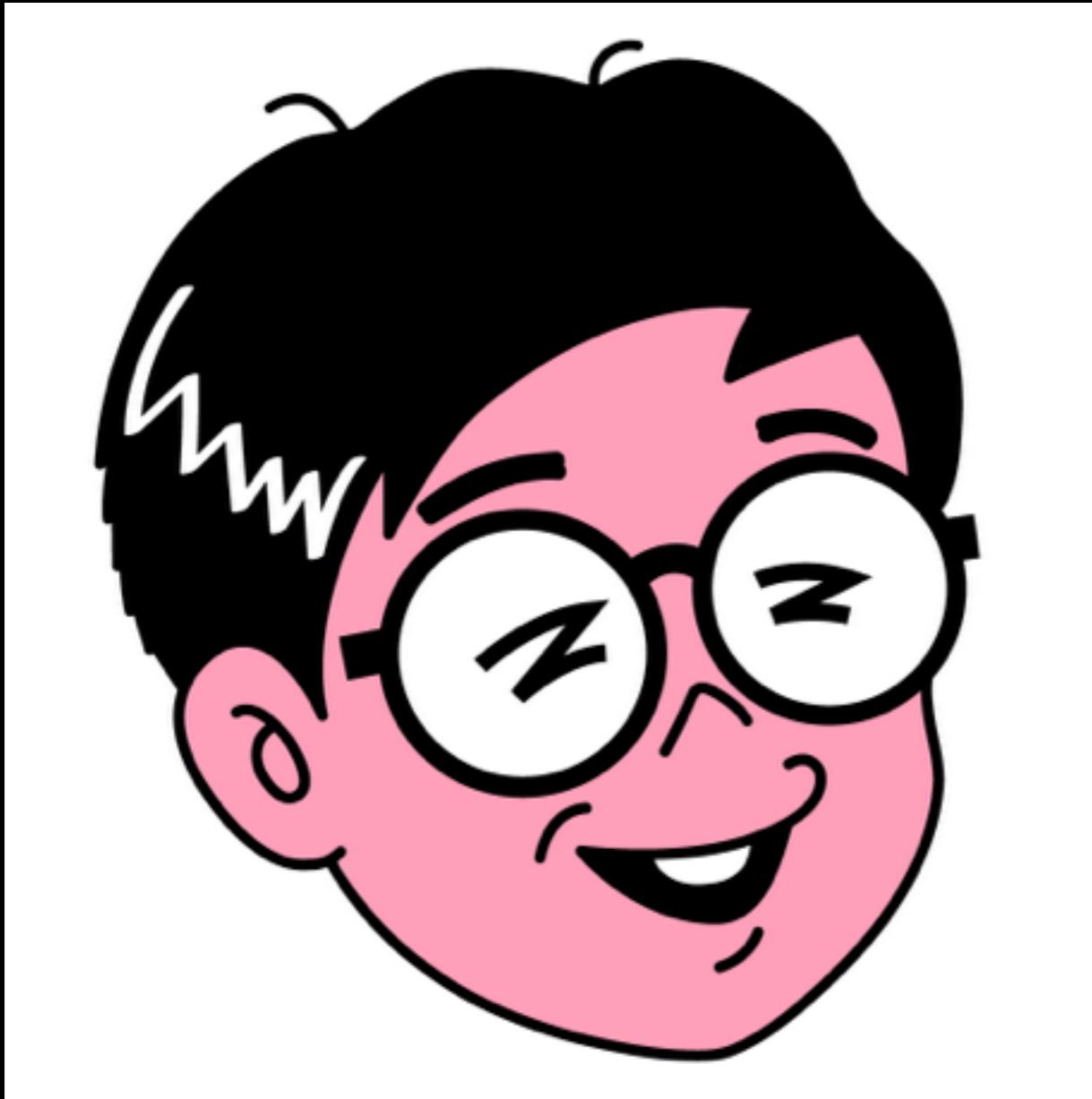


10만 라인  
26280시간

PYCON APAC 2016  
Minyoung Jeong  
The Beatpacking Company



**Respect, Diversity**  
PyCon APAC 2016



Minyoung Jeong

CTO, The Beatpacking  
Company

2014/2015 PyCON KR

Founder, AWSKRUG

AWS Community Hero

Love Python!



**Respect, Diversity**  
PyCon APAC 2016

# 제약을 넘어 - Gevent

PYCON KOREA 2014  
정민영 @ 비트패킹컴퍼니



**Respect, Diversity**  
PyCon APAC 2016

# Celery의 빛과 그림자

PYCON KOREA 2015

정민영

THE BEATPACKING COMPANY



**Respect, Diversity**  
PyCon APAC 2016

2013. 04. 04



**Respect, Diversity**  
PyCon APAC 2016

9 changed files with 787  
additions and 0 deletions



.gitignore  
README.rst  
beat/\_\_init\_\_.py  
docs/Makefile  
docs/beat.rst  
docs/conf.py  
docs/make.bat  
setup.py



.gitignore

README.rst

beat/\_init\_.py

docs/Makefile

docs/beat.rst

docs/conf.py

docs/make.bat

setup.py



.gitignore  
README.rst  
beat/\_\_init\_\_.py  
docs/Makefile  
docs/beat.rst  
docs/conf.py  
docs/make.bat  
**setup.py**



2016. 08. 10



**Respect, Diversity**  
PyCon APAC 2016



15,728 commits

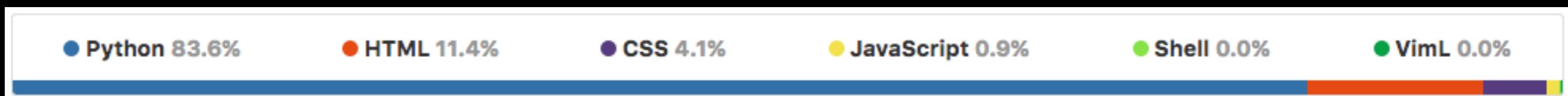
9 branches

0 releases

18 contributors



**Respect, Diversity**  
PyCon APAC 2016



```
→ beat-api git:(master) cloc .
 1560 text files.
 1517 unique files.
 119 files ignored.
```

```
github.com/AlDanial/cloc v 1.70  T=8.83 s (163.7 files/s, 32021.2 lines/s)
```

Language	files	blank	comment	code
Python	961	29900	19987	108884
HTML	367	5265	195	51093
JavaScript	50	6501	5276	32712
CSS	42	1149	1181	16771
SASS	7	283	17	1743
XML	9	0	0	1178
DOS Batch	1	29	1	212
make	1	29	6	145
YAML	2	4	0	104
Bourne Again Shell	1	14	3	66
INI	2	8	0	21
Bourne Shell	2	2	7	16
JSON	1	0	0	3
SUM:	1446	43184	26673	212948



```
→ beat-api git:(master) cloc .
 1560 text files.
 1517 unique files.
 119 files ignored.
```

github.com/AlDanial/cloc v 1.70 T=8.83 s (163.7 files/s, 32021.2 lines/s)

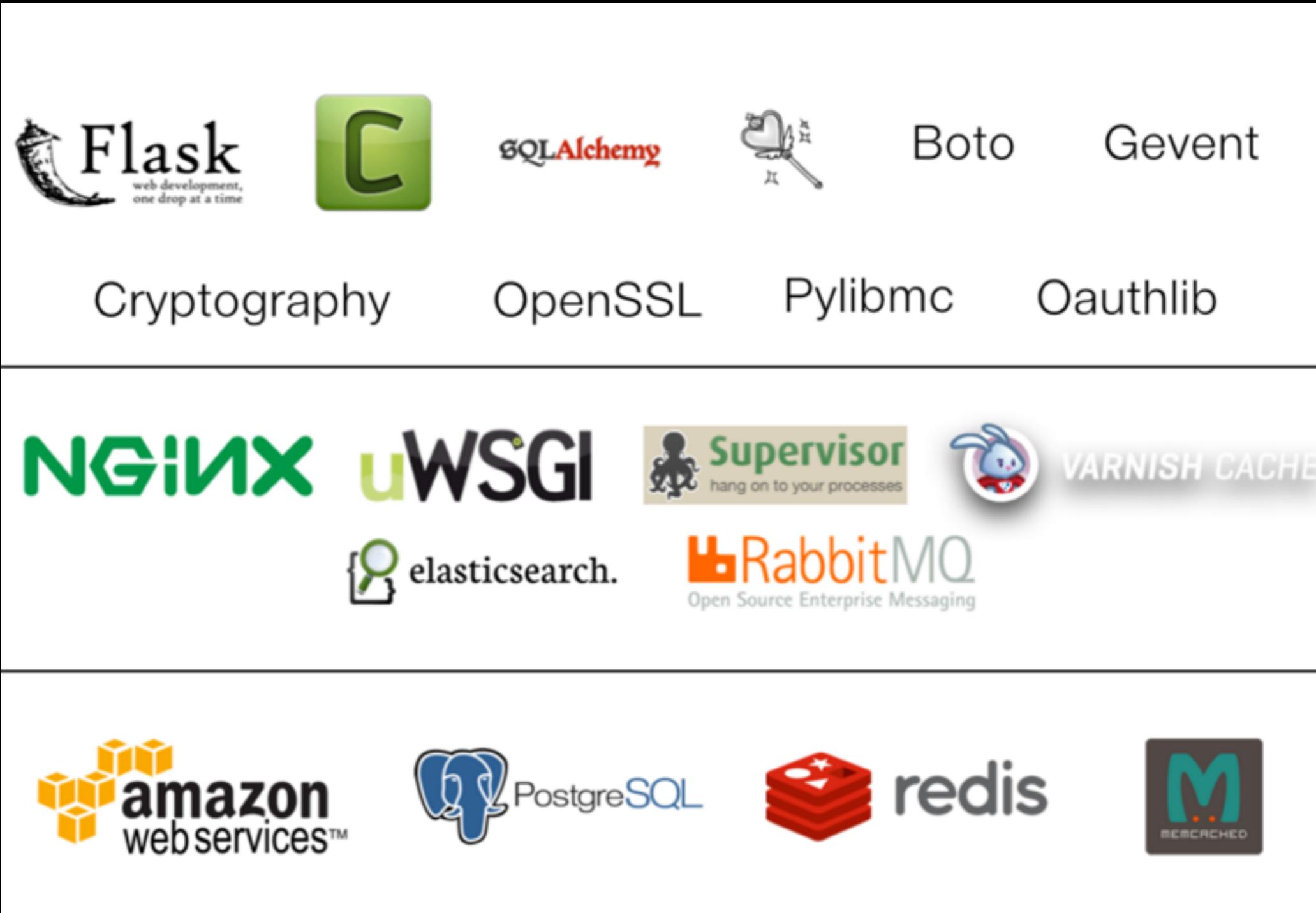
Language	files	blank	comment	code
Python	961	29900	19987	108884
HTML	367	5265	195	51093
JavaScript	50	6501	5276	32712
CSS	42	1149	1181	16771
SASS	7	283	17	1743
XML	9	0	0	1178
DOS Batch	1	29	1	212
make	1	29	6	145
YAML	2	4	0	104
Bourne Again Shell	1	14	3	66
INI	2	8	0	21
Bourne Shell	2	2	7	16
JSON	1	0	0	3
SUM:	1446	43184	26673	212948



- REST API Server
- CMS
- Backoffice
- Automation
- … Client 개발을 제외한 거의 전부

# Single monolithic repository

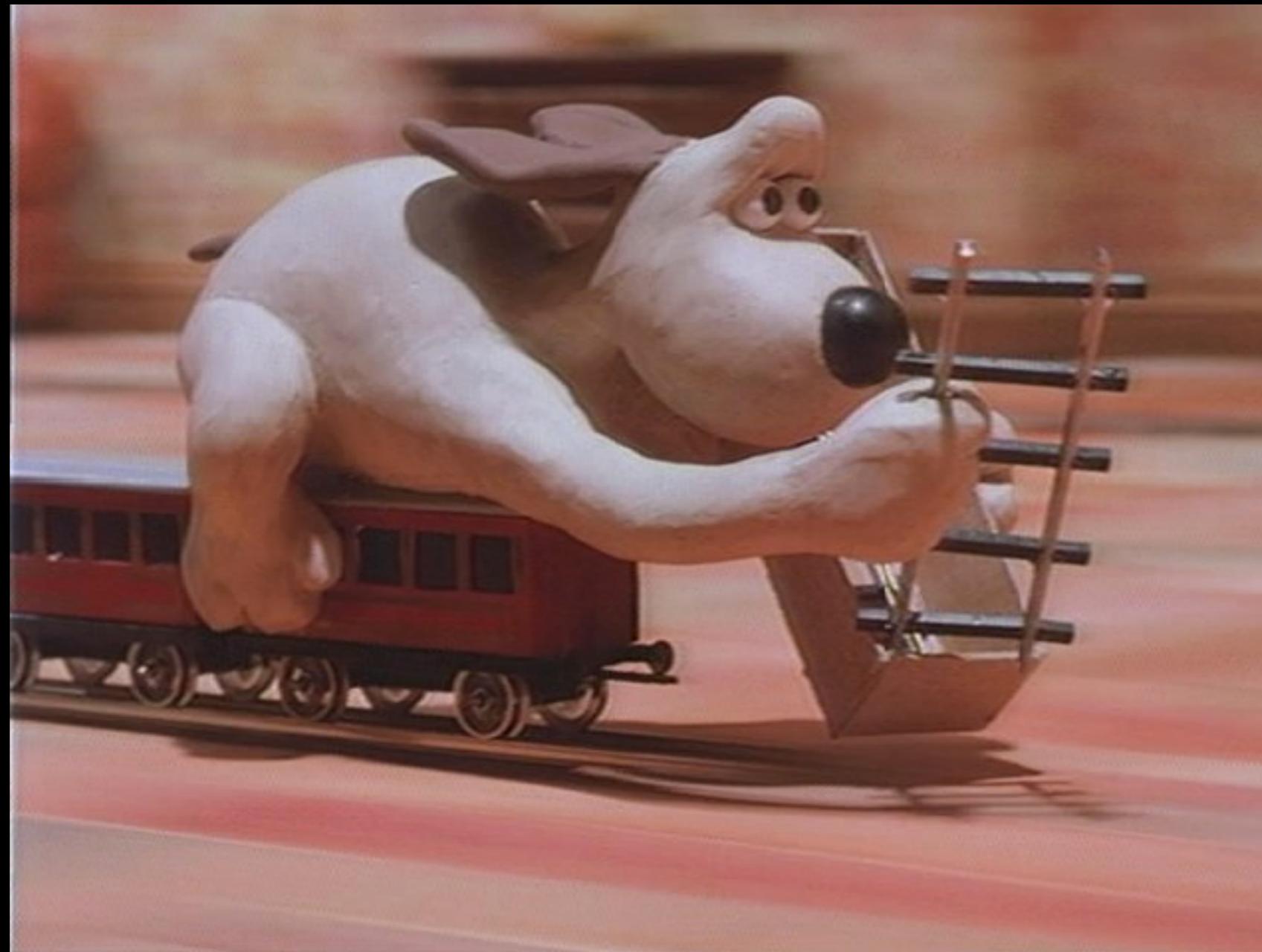




스타트업에서  
개발을 한다는건…



**Respect, Diversity**  
PyCon APAC 2016



철로가 준비되는 시간을  
기다려 주지 않는다



**Respect, Diversity**  
PyCon APAC 2016

일정을 개발이 주도하기  
어렵고 변수가 많다



**Respect, Diversity**  
PyCon APAC 2016

목표

최소한의 품질은 항상 달성하자



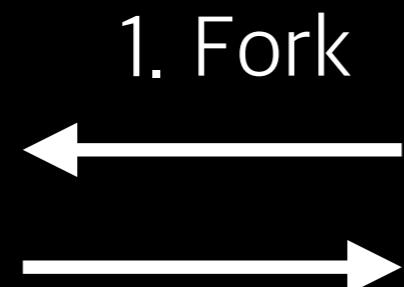
**Respect, Diversity**  
PyCon APAC 2016

# 극한 직업 비트 개발팀이 일하는 법





2. Push



3. Pull Request



7. Merge

4. Review  
& Approval



5. CI & Lint &Codecov

6. Result  
hook



**Respect, Diversity**  
PyCon APAC 2016



**Respect, Diversity**  
PyCon APAC 2016



“Github\_장애.jpg”



**Respect, Diversity**  
PyCon APAC 2016

# 과업(Task) 단위의 PR





“출시 3일전에 실행한 ‘git merge upstream master’에서  
1395개의 충돌을 발견한 프로그래머”

가능하면 작은 단위로



**Respect, Diversity**  
PyCon APAC 2016

	Author	Labels	Milestones	Assignee	Sort
<input type="checkbox"/> 7 Open	✓ 4,727 Closed				
<input type="checkbox"/> <a href="#">Access</a>	<a href="#">#5176</a>				5
<input type="checkbox"/> <a href="#">Pypy</a>	<a href="#">#4792</a>				12
<input type="checkbox"/> <a href="#">gcm</a>		✓ in progress			4
<input type="checkbox"/> <a href="#">WIP:</a>		개기 ✓ in progress	0 of 2		18
<input type="checkbox"/> <a href="#">음원</a>		✓			2
<input type="checkbox"/> <a href="#">공식 계정들의 팔로</a>			✓		6
<input type="checkbox"/> <a href="#">광고 통계 수집</a>		●			3



# Review





# Review의 이유

- 해당 구현에 대한 팀원간 이해를 높인다.
- 좀 더 나은 구현에 대한 토론을 가능하게 한다.
- 심지어는 해당 구현이 필요한지에 대한 얘기 조차도 할 수 있다.
- 인수 인계를 하게 된다(!)

diff icon  yoloseem commented on the diff 10 days ago

tests/web/memcache\_test.py No coverage [View full changes](#)

```
184 +     assert not p.called
185 +     jobs = [gevent.spawn(_x, k[0], k[1]) for k in [('aa', 1), ('bb',
186 +         gevent.joinall(jobs)
187 +         assert all([j.successful() for j in jobs])
188 +         assert p.called
189 +         assert 1 == p.call_count
190 +
191 +
192 +def test_get_wo_pipeline():
193 +    import gevent
194 +    with app.app_context():
195 +        mc = AsyncMemcacheClientProxy([
196 +            '127.0.0.1:11211',
197 +            ], get_pipeline=False, always_eager=True)
198 +
199 +    ogma = mc.get_many
```

 yoloseem added a note 10 days ago +😊 ✎ ×

(무엇의 약자인 것인가.....)

 kkung added a note 10 days ago +😊 ✎ ×

아마도 original get many..? ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ

Add a line note



# Merge 정책





<http://homu.io>



**Respect, Diversity**  
PyCon APAC 2016

## Branch protection for master

### Protect this branch

Disables force-pushes to this branch and prevents it from being deleted.

### Require status checks to pass before merging

Choose which [status checks](#) must pass before branches can be merged into master. When enabled, commits must first be pushed to another branch, then merged or pushed directly to master after status checks have passed.

### Include administrators

Enforce required status checks for repository administrators.

### Require branches to be up to date before merging

This ensures the branch has been tested with the latest code on master.

Status checks found in the last week for this repository

codecov/patch

codecov/project

continuous-integration/travis-ci

homu

Required

### Restrict who can push to this branch

Specify people or teams allowed to push to this branch. Required status checks will still prevent these people from merging if the checks fail.

[Save changes](#)



## Branch protection for master

**Protect this branch**

Disables force-pushes to this branch and prevents it from being deleted.

**Require status checks to pass before merging**

Choose which [status checks](#) must pass before branches can be merged into master. When enabled, commits must first be pushed to another branch, then merged or pushed directly to master after status checks have passed.

**Include administrators**

Enforce required status checks for repository administrators.

**Require branches to be up to date before merging**

This ensures the branch has been tested with the latest code on master.

Status checks found in the last week for this repository

codecov/patch

codecov/project

continuous-integration/travis-ci

homu

Required

**Restrict who can push to this branch**

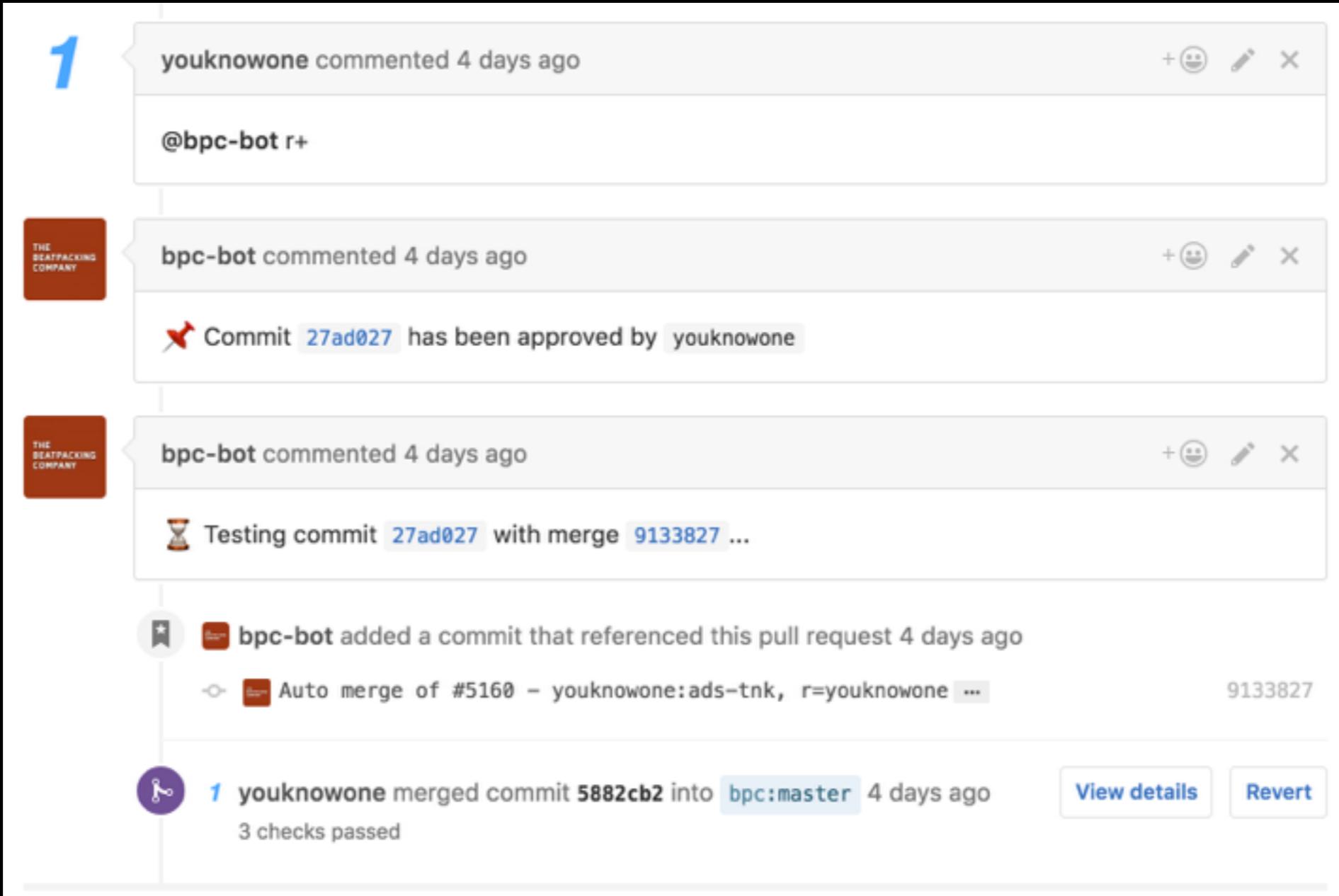
Specify people or teams allowed to push to this branch. Required status checks will still prevent these people from merging if the checks fail.

**Save changes**



# 안전한 MASTER

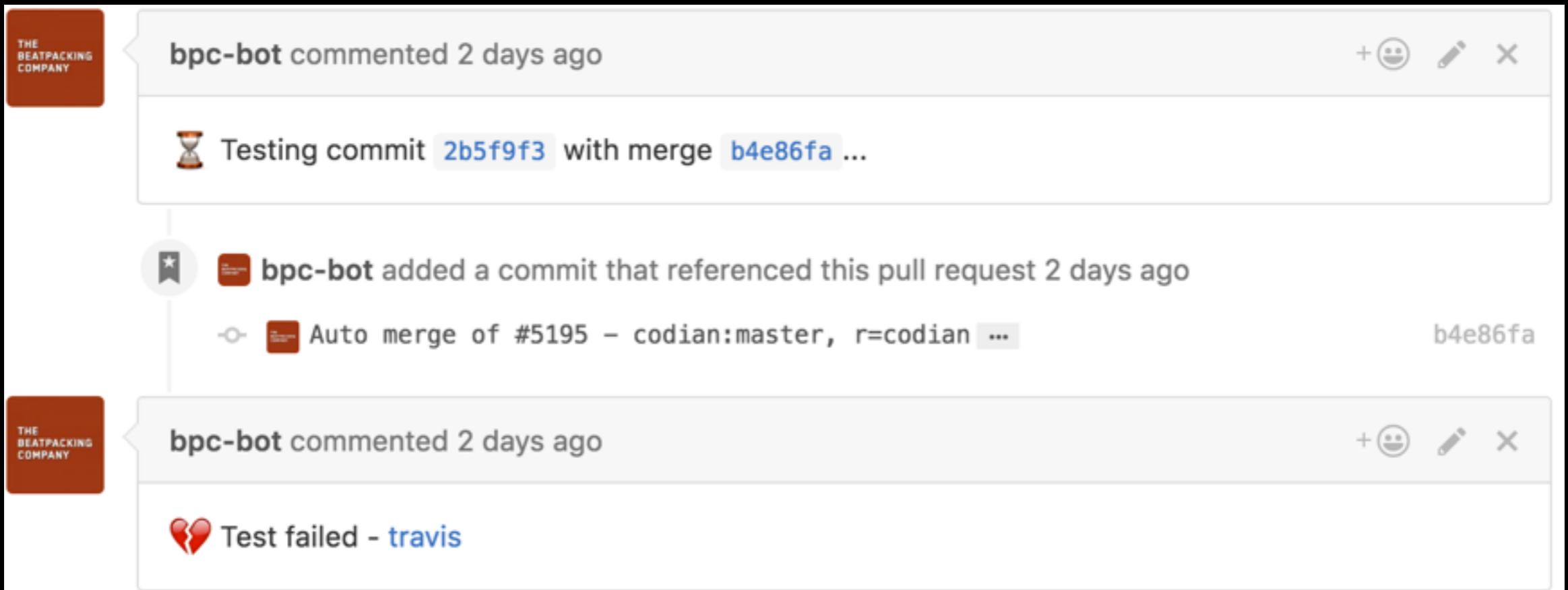




The screenshot shows a sequence of events in a GitHub pull request timeline:

- youknowone commented 4 days ago**  
@bpc-bot r+
- bpc-bot commented 4 days ago**  
Commit [27ad027](#) has been approved by youknowone
- bpc-bot commented 4 days ago**  
Testing commit [27ad027](#) with merge [9133827](#) ...
- bpc-bot added a commit that referenced this pull request 4 days ago**
  - Auto merge of #5160 – youknowone:ads-tnk, r=youknowone ... [9133827](#)
- youknowone merged commit [5882cb2](#) into [bpc:master](#) 4 days ago**  
3 checks passed





The screenshot shows two GitHub pull request comment threads from a bot named bpc-bot.

**Top Thread:**

- bpc-bot commented 2 days ago**
-  Testing commit [2b5f9f3](#) with merge [b4e86fa](#) ...

**Bottom Thread:**

-   bpc-bot added a commit that referenced this pull request 2 days ago
  -  Auto merge of #5195 – codian:master, r=codian ... b4e86fa
- bpc-bot commented 2 days ago**
-  Test failed - travis



The screenshot shows a GitHub pull request comment. On the left, there's a small orange square icon with the text "THE BEATPACKING COMPANY". The main area contains a comment from "bpc-bot" that was posted 8 days ago. The comment text is: "⚠️ The latest upstream changes (presumably [#5134](#)) made this pull request unmergeable. Please resolve the merge conflicts." To the right of the comment, there are three icons: a plus sign with a smiley face, a pencil, and a close button.

bpc-bot commented 8 days ago

⚠️ The latest upstream changes (presumably [#5134](#)) made this pull request unmergeable. Please resolve the merge conflicts.



테스트



**Respect, Diversity**  
PyCon APAC 2016



# 무엇을 점검할까?

- 사람은 항상 실수한다. 실수를 blame하지 말고 실수를 하지 못하게 하자.
- 프로젝트가 시간이 지나면 지날수록 코드의 가독성은 바로 생산성이 된다.
- 테스트는 외부에 노출되는 Endpoint를 주된 목표로 하자.





# lint - 개발 권고 사항 강제



# lint - import 실수 예방



# PEP 8 -- Style Guide for Python Code

PEP:	8
Title:	Style Guide for Python Code
Author:	Guido van Rossum <guido at python.org>, Barry Warsaw <barry at python.org>, Nick Coghlan <ncoghlan at gmail.com>
Status:	Active



# flake8 / autopep8



E501,F401,E402,F841



**Respect, Diversity**  
PyCon APAC 2016



**Respect, Diversity**  
PyCon APAC 2016

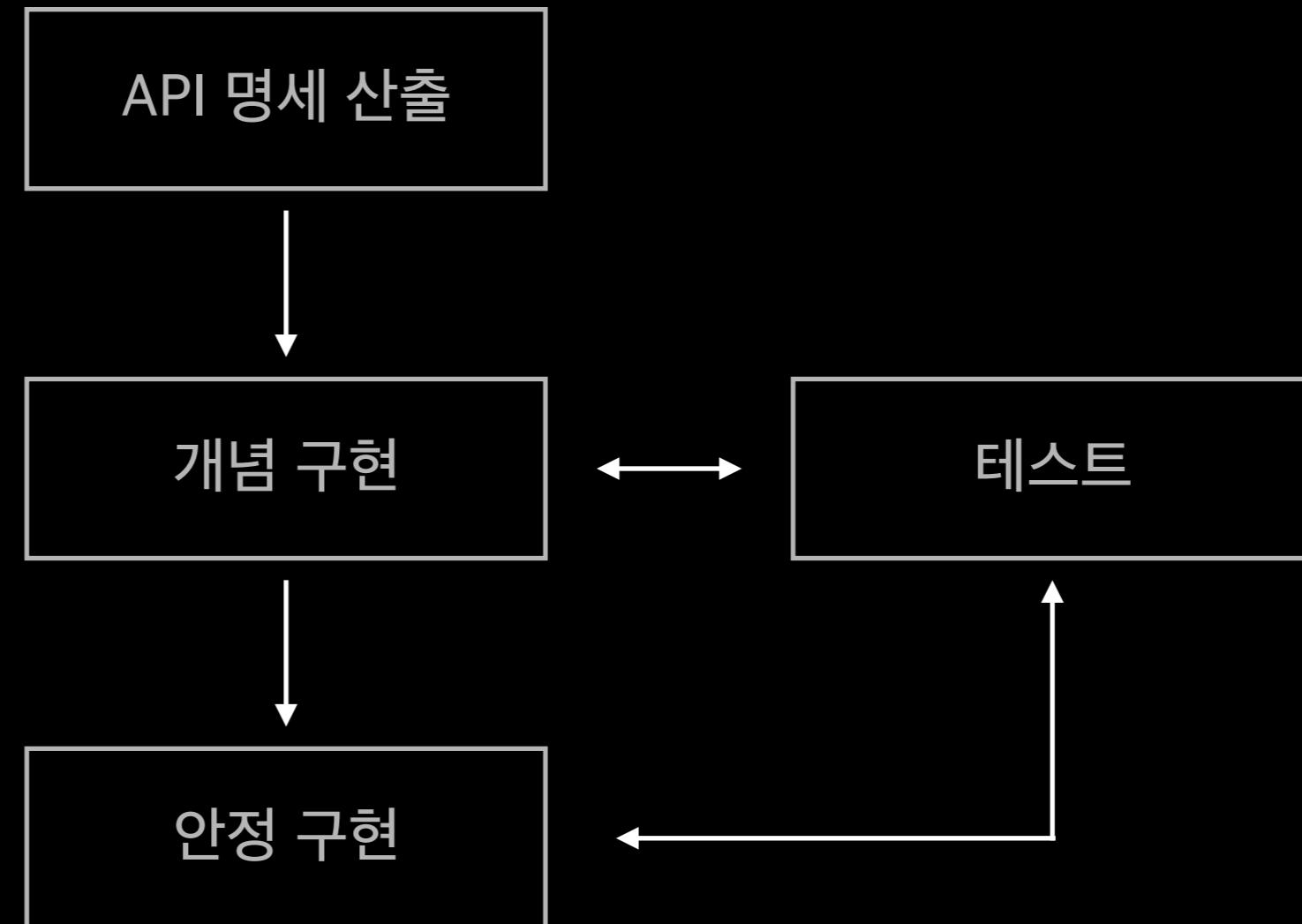
# 왜 Endpoint 위주로 테스트를 할까?



# 미성숙한 요구사항 API 위주의 개발



**Respect, Diversity**  
PyCon APAC 2016



- REST Endpoint를 호출하여 의도된 결과와 의도된 실패를 점검한다.
- 자명한 것 ( $1+1 = 2$ )을 테스트 한다. Framework나 의존성의 업그레이드시 발생하는 Breaking Changes에 대응할 수 있다.
- 외부 I/O에 대한 상태를 관리할 수 있다면 Mock보다는 실제 서버를 이용한다. (버그를 피하거나, 버그를 전제로 하거나.)
- 테스트에서 놓쳤던 버그는 테스트를 꼭 만들어 넣어야. 자존감을 유지할 수 있다.

# 빠른 테스트는 매우 중요



**Respect, Diversity**  
PyCon APAC 2016

테스트가 느리면  
CI를 장애물로 여기게 된다



# test coverage



▼ Showing 1 of 1 changed files with 21.43% of changed lines covered. [View details](#)

beat/async.py changed.

3 0 11 21.43%

► Untracked and deleted files

beat / async.py		27 3 0 11 21.43%
@@ -13,4 +13,29 @@		
13	13	
14	14	
15	15	
16	- logging.info('ENABLE_ASYNC: %r' % enable_async)	
16	+ logger = logging.getLogger('beat.async')	
17	+ logger.info('ENABLE_ASYNC: %r' % enable_async)	
18	+ if enable_async:	
19	+ import greenlet	
20	+ import time	
21	+ import gevent	
22	+ import traceback	
23	+ MAX_BLOCKING_TIME = 0.5 # 500ms	
24	+ _last_switch_time = None	
25	+ def _tracer(what, (origin, target)):	
26	+     global _last_switch_time	
27	+     then = _last_switch_time	
28	+     now = _last_switch_time = time.time()	
29	+	
30	+     if then is not None:	
31	+         blocking_time = now - then	
32	+         if blocking_time > MAX_BLOCKING_TIME and origin is not gevent.get_hub():	
33	+             tb = ''.join(traceback.format_stack(origin.gr_frame))	
34	+             logger.warn('Greenlet blocked more than %.4f seconds\n%s' % (blocking_time, tb))	
35	+	
36	+     greenlet.settrace(_tracer)	
@@ -17 +42 @@		

# 만든 테스트의 유효성



이 PR의 CI 통과를  
얼마나 신뢰할 수 있는가?



 All checks have passed [Hide all checks](#)

3 successful checks

	<b>codecov/patch</b> — 87.50% of diff hit (target 51.83%)	<a href="#">Details</a>
	<b>codecov/project</b> — 51.83% (+<.01%) compared to a056751	<a href="#">Details</a>



# 문서



“실수로 문서화를 영어로 했었음”

*–Hong Minhee , creator of Wand*



**Respect, Diversity**  
PyCon APAC 2016

# 실수로 문서화를 영어로 했었음

⌚ master ⏰ deploy-1.30.157 ... build-65



Hong Minhee ? committed on 3 May 2013



Respect, Diversity  
PyCon APAC 2016

# Sphinx + httpdomain



```
@bp.route('/<int:followee_id>/followers/<int:follower_id>/')
def follower(followee_id, follower_id):
    """팔로잉 정보를 표시한다. :http:method:`HEAD` 요청으로
    팔로잉 관계가 있는지 확인하는 용도로 쓸 수 있다.

    .. code-block:: http

        HEAD           /followers/2/ HTTP/1.1
        Accept: application/json
        Host:   acking.com
        User-Agent: BEAT-Android/1.0.0

    팔로잉 관계가 있을 경우 :http:statuscode:`200` 응답이 온다.

    .. code-block:: http

        HTTPS/1.0 200 OK
        Content-Type: application/json
        Date: Mon, 15 Apr 2013 13:23:04 GMT
        Server: BEAT-API/1.0.0

    팔로잉 관계가 없으면 :http:statuscode:`404` 응답이 온다.

    .. code-block:: http

        HTTPS/1.0 404 Not Found
        Content-Type: application/json
        Date: Mon, 15 Apr 2013 13:23:04 GMT
        Server: BEAT-API/1.0.0

    :param followee_id: :attr:`Following.followee_id`
                        <beat.user.Following.followee_id>
    :type followee_id: :class:`numbers.Integral`
    :param follower_id: :attr:`Following.follower_id`
                        <beat.user.Following.follower_id>
    :type follower_id: :class:`numbers.Integral`
    :status 200: 팔로잉 관계가 있는 경우
    :status 404: 팔로잉 관계가 없는 경우
    ....
```





코드에서 문서를 찾는 개발자들.jpg

```
    track, track, voice_caption=None,  
    ...  
:param track:  
:param voice_caption:  
:param curated_by:  
:param episode_leaf: 에피소드의 마지막 트랙인지  
:type episode_leaf: :class:`boolean`  
:param next_episode: 다음 에피소드 번호  
:return:  
    ...
```



# 문서화에 실패한 이유

- 내부 고객(클라이언트 개발자)과의 커뮤니케이션에 있어서 주요한 수단이 되지 못했다.
- 문서를 보는 사람이 없으니 간신이 되질 않아 문서 자체의 효용이 사라짐.
- 문서 자체의 효용이 사라져 다시 주된 수단이 되기 어려워 지는 악순환의 반복.





사업을 위해 부채를 피할  
수 없는 것 처럼....



**Respect, Diversity**  
PyCon APAC 2016

적절한 기술 부채는  
꼭 필요하다



**Respect, Diversity**  
PyCon APAC 2016

**"Today we're going to add  
a third floor to our house!"**



Sources: <http://www.flickr.com/photos/spleenness/8222224998>  
<http://www.flickr.com/photos/perspective/5662798485>

# 적정 기술



# 기술을 적용함에 있어 적정한 시점과 비용에 대한 고민



오늘도 기술부채를 늘린  
개발자가 용서를 구하고 있다.jpg



Respect, Diversity  
PyCon APAC 2016

# 사례1. 비용 평가의 실패



# 2013년

- `async` 좋은건 알지만….
- 도입하면 복잡도가 증가하고 (아직) 필요가 크지 않다.
- 나중에 필요하면 도입할 수 있지 않겠어…?

# 그리고 도래한 2016년



**Respect, Diversity**  
PyCon APAC 2016

# Async가 필요하다!

- 인스턴스당 처리 효율이 너무 떨어져 비용이 낭비된다.
- 외부 시스템 연동이 이제 너무 많아서 대부분 시간이 I/O
- gevent monkeypatch가 모든걸 해결해 주시겠지…?



그때와 지금은  
모든게 다르다



**Respect, Diversity**  
PyCon APAC 2016



# 시점에 대한 후회

- 2013년에 단 3줄로 처리 되었을 문제가….
- 2016년엔 2000줄의 수정과 31개의 파일 변경, 개발+테스트+머지까지 1달 이상의 기간을 필요로 했다.
- 이미 복잡도가 상승한 시스템에서 ‘monkey patch’같은 magic은 불가능.
- 개발 초기 단기의 복잡도 상승을 과대 평가하여 이후 막대한 비용을 지불.



# 사례2





*Read The F\*\*\* Manual*



Respect, Diversity  
PyCon APAC 2016

# SQLAlchemy Transaction Management



## Autocommit Mode

The example of `Session` transaction lifecycle illustrated at the start of [Managing Transactions](#) applies to a `Session` configured in the default mode of `autocommit=False`. Constructing a `Session` with `autocommit=True` produces a `Session` placed into “autocommit” mode, where each SQL statement invoked by a `Session.query()` or `Session.execute()` occurs using a new connection from the connection pool, discarding it after results have been iterated. The `Session.flush()` operation still occurs within the scope of a single transaction, though this transaction is closed out after the `Session.flush()` operation completes.

### Warning

“autocommit” mode should **not be considered for general use**. If used, it should always be combined with the usage of `Session.begin()` and `Session.commit()`, to ensure a transaction demarcation.

Executing queries outside of a demarcated transaction is a legacy mode of usage, and can in some cases lead to concurrent connection checkouts.

In the absence of a demarcated transaction, the `Session` cannot make appropriate decisions as to when autoflush should occur nor when auto-expiration should occur, so these features should be disabled with `autoflush=False, expire_on_commit=False`.



*“autocommit” mode  
should not be considered  
for general use.*



# Implicit vs Explicit



# Implicit Transaction

- SQLAlchemy가 암시적 Transaction 관리를 수행한다.
- 기본적으로 session이 생성되는 시점에 무조건 Transaction을 수행하고, commit 혹은 rollback 되어 종료되는 즉시 새로운 Transaction 생성.
- INSERT/UPDATE/DELETE같은 변경뿐 아니라 SELECT만 수행하는 경우에도 Transaction 문맥하에서 수행된다.



14.010254/34836833].

 **Dr. House** BOT 6:54 AM  
BEAT-API is down (Incident #419)  
[beat-api.beatpacking.com](http://beat-api.beatpacking.com) • View details

 **Dr. House** BOT 7:00 AM  
BEAT-API is up (Incident #419)  
[beat-api.beatpacking.com](http://beat-api.beatpacking.com) • View details

 **Dr. House** BOT 7:06 AM  
BEAT-API is down (Incident #420)  
[beat-api.beatpacking.com](http://beat-api.beatpacking.com) • View details

 **Dr. House** BOT 7:22 AM  
BEAT-API is up (Incident #420)  
[beat-api.beatpacking.com](http://beat-api.beatpacking.com) • View details

 꽃띠양 7:32 AM  
[@youknowone](#) 장애  
[@kkung](#) 장애 (edited)

 **Dr. House** BOT 7:33 AM  
BEAT-API is down (Incident #422)  
[beat-api.beatpacking.com](http://beat-api.beatpacking.com) • View details



# The Last “SPOF”

- 대부분의 DBMS에서 Transaction과 Connection 비용은 클 수 밖에 없다.
- W:R Ratio가 9:1에 가까움에도 Transaction이 과다해 DB에 부하가 너무 크게 집중되는 현상의 반복.
- Connection pooling 같은 방법이 실질적으로 효과를 발휘할 수 없다.



*“Explicit is better than implicit”*

*PEP 20 -- The Zen of Python*



Respect, Diversity  
PyCon APAC 2016

```
sessionmaker(  
    bind=engine, autocommit=True  
)
```





**FAIL**



**Respect, Diversity**  
PyCon APAC 2016

# Python의 배신

- PEP 249 — Python Database API Specification v2.0
- There isn't even a way to explicitly start a transaction in the PEP 249 API; no begin() or similar. Instead, you are always in a transaction;
- psycopg2는 PEP249의 작동을 준수하고, SQLAlchemy +psycopg2에서 autocommit mode의 설정은 작동에 영향을 주지 못한다.
- 대신 isolation\_level='AUTOCOMMIT' 으로 조정해야 한다.



문서에 없는데…?



Respect, Diversity  
PyCon APAC 2016

신속하게,  
그러나 신중하게



**Respect, Diversity**  
PyCon APAC 2016

- Session 관리는 가장 초창기에 들어간 코드 중에 하나지만, 전반적인 Scale-up에 있어 끝까지 발목을 붙잡았다.
- 일반적인 서비스에서 DB가 병목이면서 동시에 SPOF가 되는 걸 충분히 이해하고 있음에도 해당 비용을 평가하지 않았다.
- 이미 복잡도가 상승한 시점에서 Session의 설정을 바꾸는건 무척이나 비용이 크고 조심스러운 작업이 되어 쉽게 시도하지 못했다.
- 충분히 예측 가능한 기본적인 부분들에 대해서는 조금 더 신중했어야.

# 느슨하지만 엄격하게

- 처음부터 완벽한 선택을 할 수는 없기에 거기에 너무 매몰되어 적정 시점을 놓치는건 너무 뼈아프다.
- 하지만 신중해야 되는 부분에서는 신중할 필요가 있다.
- 이런 알면서 (잘)못하는 것을 어떻게 다루는지가 결국 장기적인 프로젝트 관리
- 목표는 엄격하지만, 단계는 느슨하게.

