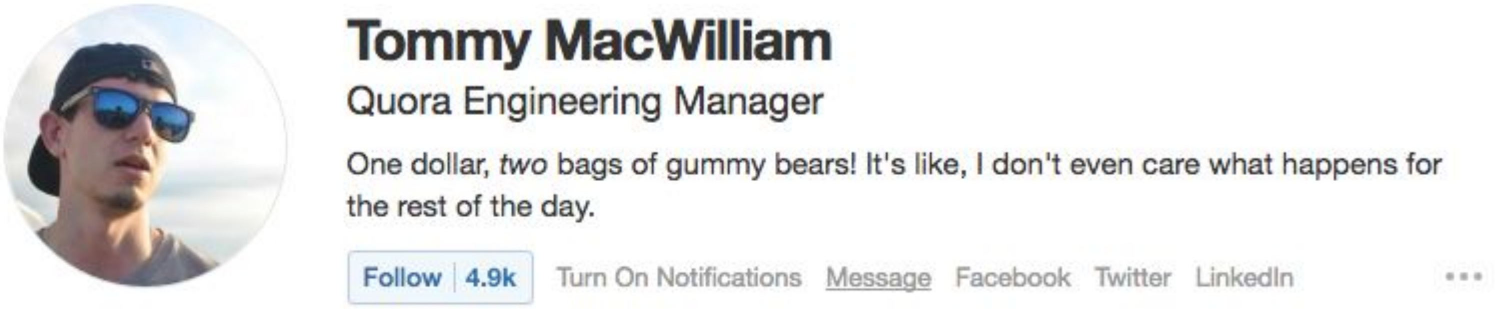
Asynchronous Programming at Quora

Quora

Background



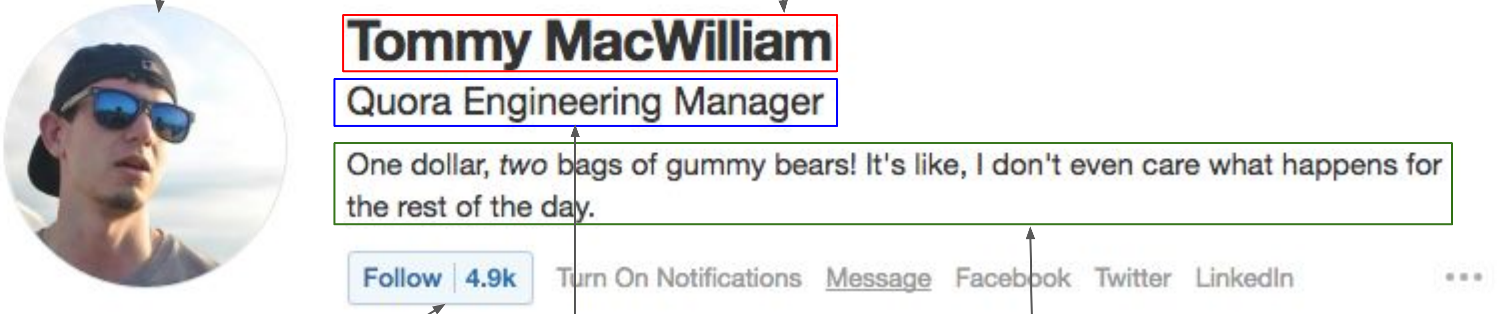
Tommy MacWilliam

Quora Engineering Manager

One dollar, *two* bags of gummy bears! It's like, I don't even care what happens for the rest of the day.

Image URL

Name



Tommy MacWilliam

Quora Engineering Manager

One dollar, *two* bags of gummy bears! It's like, I don't even care what happens for the rest of the day.

Short Bio

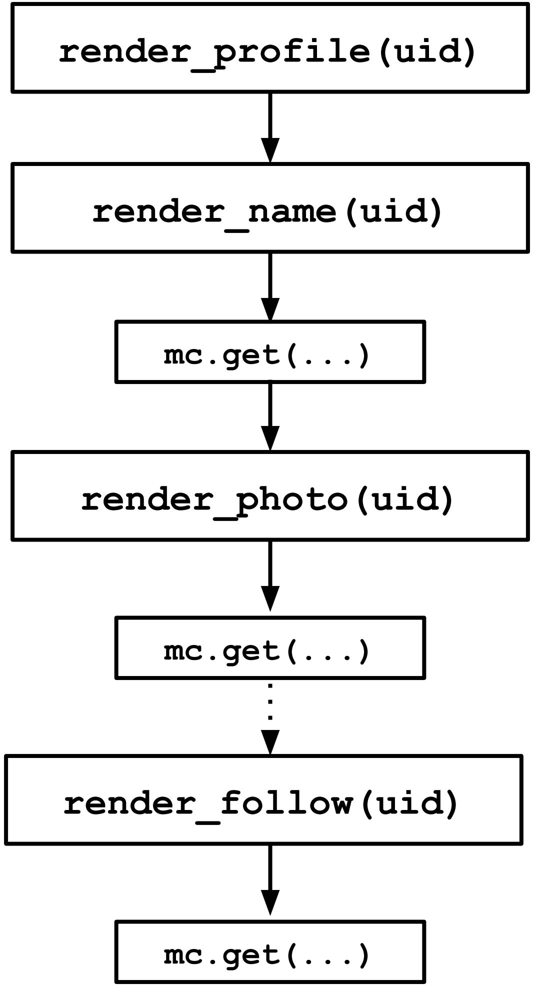
Long Bio

Follower Count

```python
def render_profile(uid):
    name = render_name(uid)
    photo = render_profile_photo(uid)
    short_bio = render_short_bio(uid)
    long_bio = render_long_bio(uid)
    follow_button = render_follow_button(uid)
    ...
```

```
def render_name(uid):
    name = mc.get(‘user-name:’ + uid)
    return ‘<b>’ + name + ‘</b>’
```

```
def render_name(uid):
    name = mc.get(‘user-name:’ + uid)
    return ‘<b>’ + name + ‘</b>’
```



render_profile(uid)

render_name(uid)

mc.get(...)

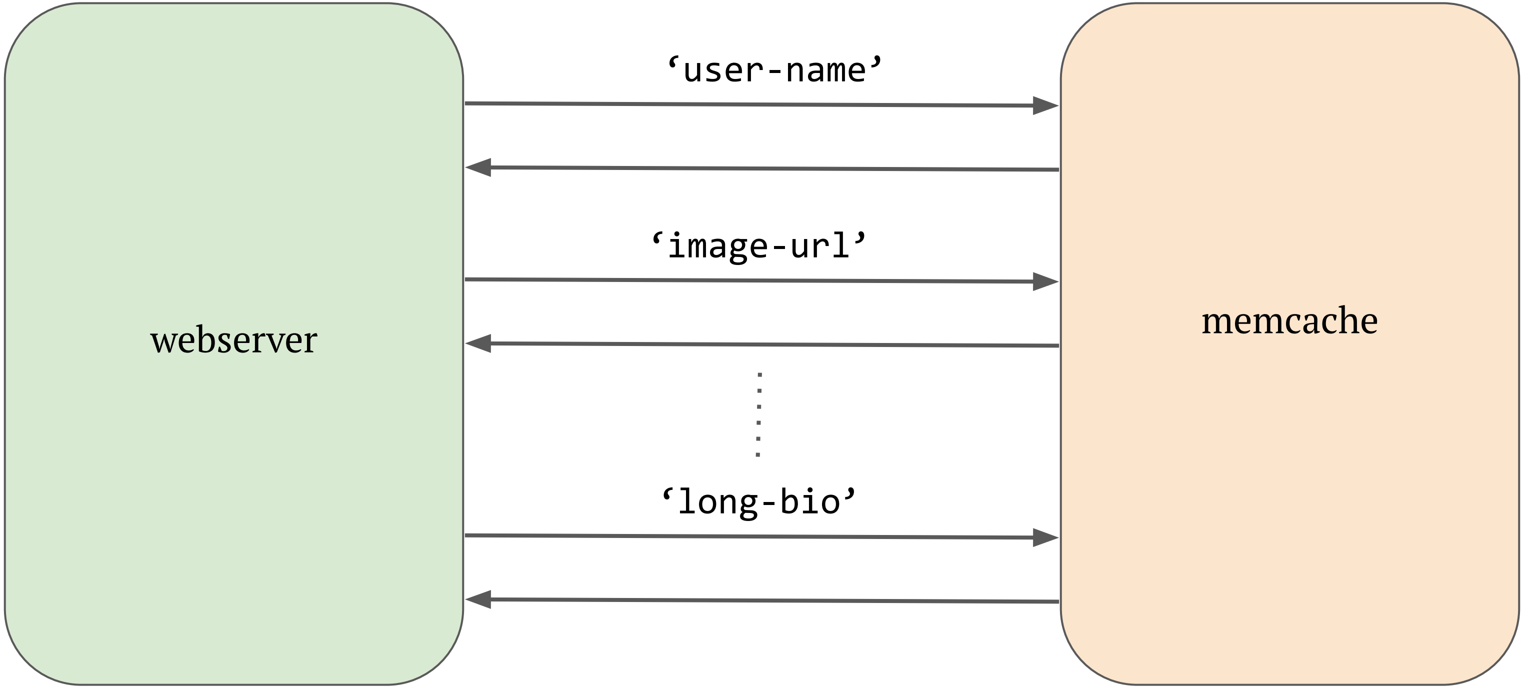
render_photo(uid)

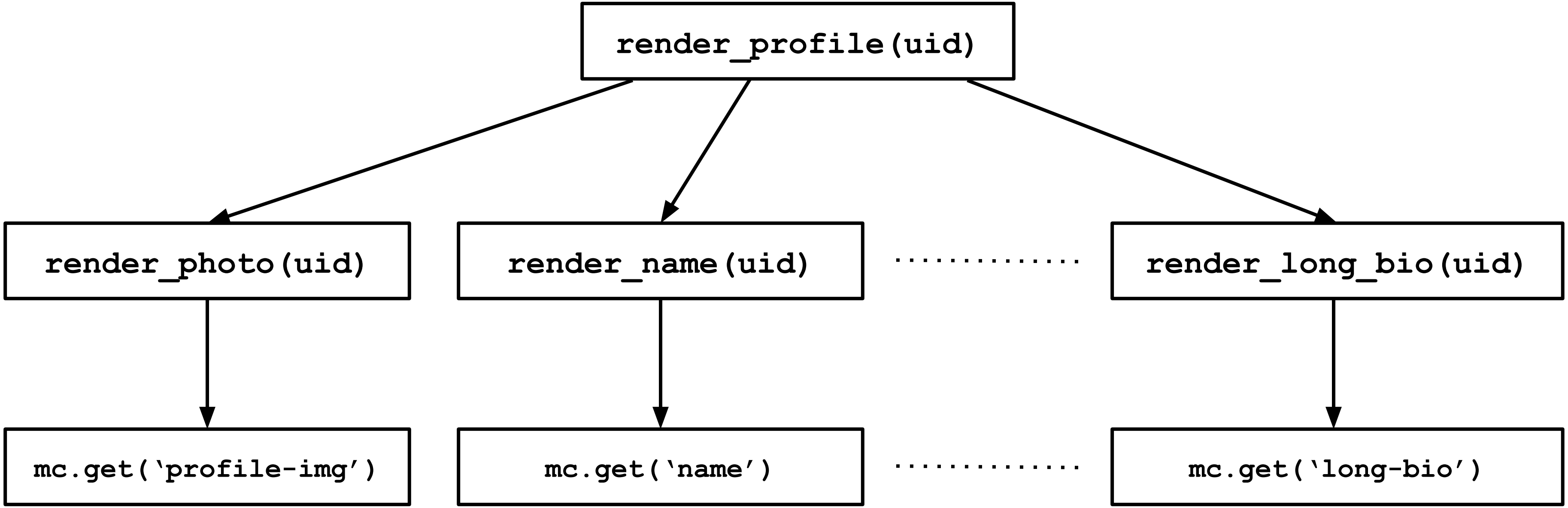
mc.get(...)

render_follow(uid)

mc.get(...)

Serial gets





render_profile(uid)

render_photo(uid)

render_name(uid)

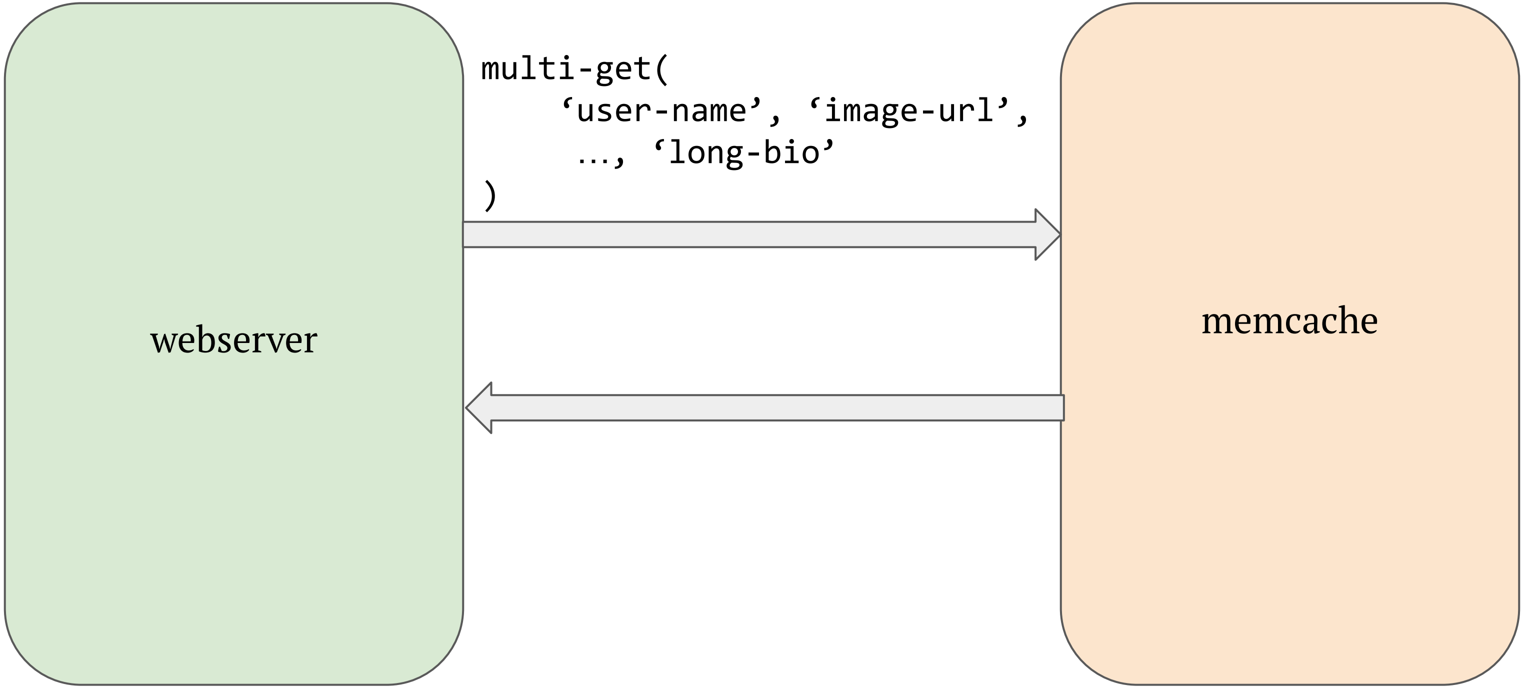
render_long_bio(uid)

mc.get('profile-img')

mc.get('name')

mc.get('long-bio')

Batched get



```
multi-get(
    'user-name', 'image-url',
     ..., 'long-bio'
)
```

Back to the code

```
def render_name(uid):
   name = mc.get(‘user-name:’ + uid)
   return ‘<b>’ + name + ‘</b>’
```

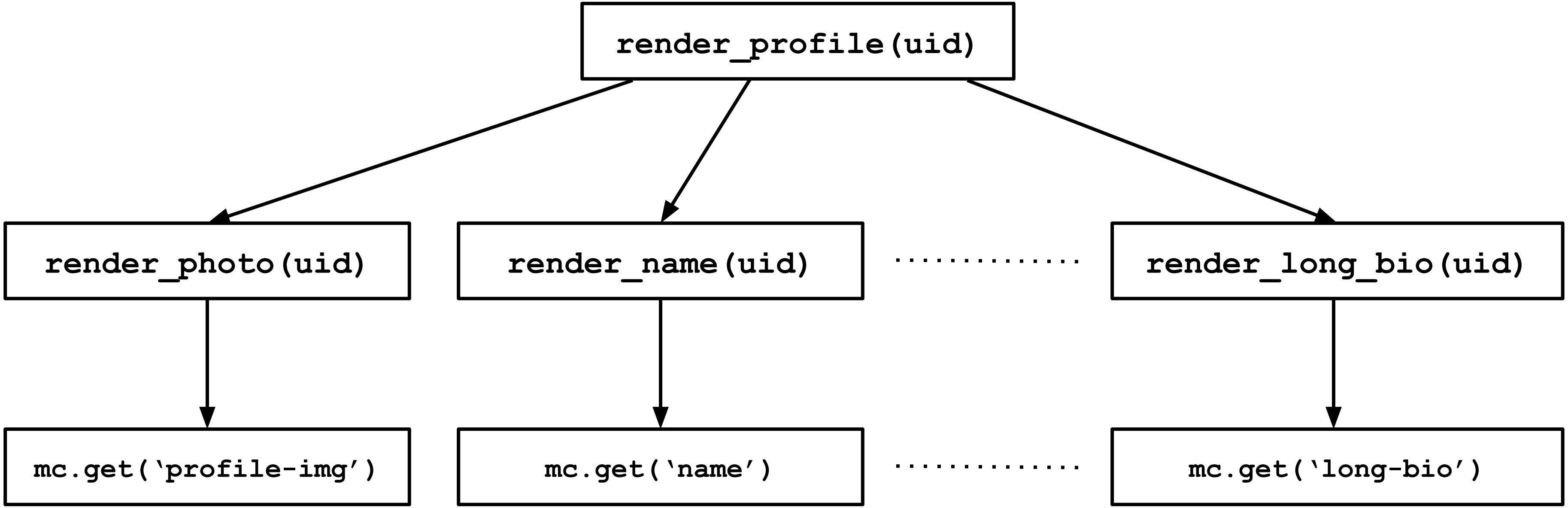
```
@async()
def render_name(uid):
    name = yield MCGet(‘user-name:’ + uid)
    return ‘<b>’ + name + ‘</b>’
```

```
@async()
def render_name(uid):
   name = yield Future(‘user-name:’ + uid)
   return ‘<b>’ + name + ‘</b>’

def scheduler():
   gen = render_name(uid)
   obj = gen.next()
   val = mc.get(obj.key)
   gen.send(val)
```

```python
def render_profile(uid):
    name = render_name(uid)
    photo = render_profile_photo(uid)
    short_bio = render_short_bio(uid)
    long_bio = render_long_bio(uid)
    follow_button = render_follow_button(uid)
    ...
```

```
@async()
def render_profile(uid):
   (photo, name, short_bio,
    long_bio, follow_button) = yield (
      render_profile_photo.async(uid),
      render_name.async(uid),
      render_short_bio.async(uid),
      render_long_bio.async(uid),
      render_follow_button.async(uid),
   )
```



render_profile(uid)

render_photo(uid)

render_name(uid)

render_long_bio(uid)

mc.get('profile-img')

mc.get('name')

mc.get('long-bio')

Thoughts on asynq

How do you migrate?

How do you migrate?

```
yield f.async() ⇔ f()
```

Does this support asynchronous I/O?

Does this support asynchronous I/O?

No, we don’t need it.

Assuming:

- Typical batch has 100 memcache gets
- Multiget is not any slower than single get

Batching gives 99% of the gains you could get from real async I/O.

Why don’t you just use asyncio?

Why don’t you just use asyncio?

- Started out on Python 2
- Asyncio isn’t designed for batching
  - But it’s easy to add
- API wouldn’t be exactly what we’re used to

How do you catch mistakes?

How do you catch mistakes?

```
nobatch.py:13:  Unnecessary yield: x was
not used before this yield
   @async()
   def h():
 -      x = yield f.async()
 -      y = yield g.async()
 +      x, y = yield f.async(), g.async()
       return x + y
```

Learn More ...

- https://engineering.quora.com/Asynchronous-Programming-in-Python
- https://github.com/quora/asynq