KU LEUVEN

Asynchronous Pub/Sub across networked processes and threads

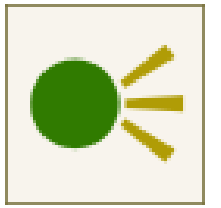
Coming soon to ManGO

Paul Borgermans

Publish / subscribe pattern

- Decoupled code
  - Publisher: (“user_added”, {“user_name”: user_name})

  - Listener: “ (“send_welcome_package”, user_name)
- Inspired on Blinker (used in ManGO Portal)
- Blinker
  - Synchronous
  - Single process
  - Inter thread “communication”
  - ~~Inter-process~~
  - ~~Over-the-wire~~



Klinker

- Publish subscribe asynchronous (world) (inter-process)
- Leverages Kumbu messaging (core of Celery)
- Many brokers
  - Redis
  - RabbitMQ
  - ..
- 150 lines of Python code
- As simple to use as Blinker

```python
ns = Namespace(BROKER_URL)
sig = ns.signal("user_updates", fanout=True)

@sig.connect
def on_user_update(sender, **kwargs):
    print(f"[{listener_name}] received from '{sender}': {kwargs}")
```

Use cases in the iRODS / ManGO ecosystem

- Sandboxed ingest workflows
  - Staging from untrusted sources
- Egress workflows
  - Data should be pulled by untrusted sources
- ManGO Portal
  - Hooks for plugins and extensions that work across containers
  - Updating in memory/session data
    - Changing group memberships
    - Access rights
- Interplanetary messaging and orchestration

Klinker

```
❯ python docs/examples/fanout_listener.py listener-A
[listener-A] listening on 'user_updates' (fanout)... Ctrl+C to stop.
[listener-A] received from 'group_manager': {'message': 'user update u6606', 'user': 'u6606'}
[listener-A] received from 'group_manager': {'message': 'user update u6856', 'user': 'u6856'}
[listener-A] received from 'group_manager': {'message': 'user update u2788', 'user': 'u2788'}
[listener-A] received from 'group_manager': {'message': 'user update u8783', 'user': 'u8783'}
[listener-A] received from 'group_manager': {'message': 'user update u7613', 'user': 'u7613'}
```

```
❯ python docs/examples/fanout_publisher.py
[publisher] sending: user update u6606
[publisher] sending: user update u6856
[publisher] sending: user update u2788
[publisher] sending: user update u8783
[publisher] sending: user update u7613
[publisher] done.
```

Kombu broker/messaging

```
❯ python docs/examples/fanout_listener.py listener-B
[listener-B] listening on 'user_updates' (fanout)... Ctrl+C to stop.
[listener-B] received from 'group_manager': {'message': 'user update u6606', 'user': 'u6606'}
[listener-B] received from 'group_manager': {'message': 'user update u6856', 'user': 'u6856'}
[listener-B] received from 'group_manager': {'message': 'user update u2788', 'user': 'u2788'}
[listener-B] received from 'group_manager': {'message': 'user update u8783', 'user': 'u8783'}
[listener-B] received from 'group_manager': {'message': 'user update u7613', 'user': 'u7613'}
```