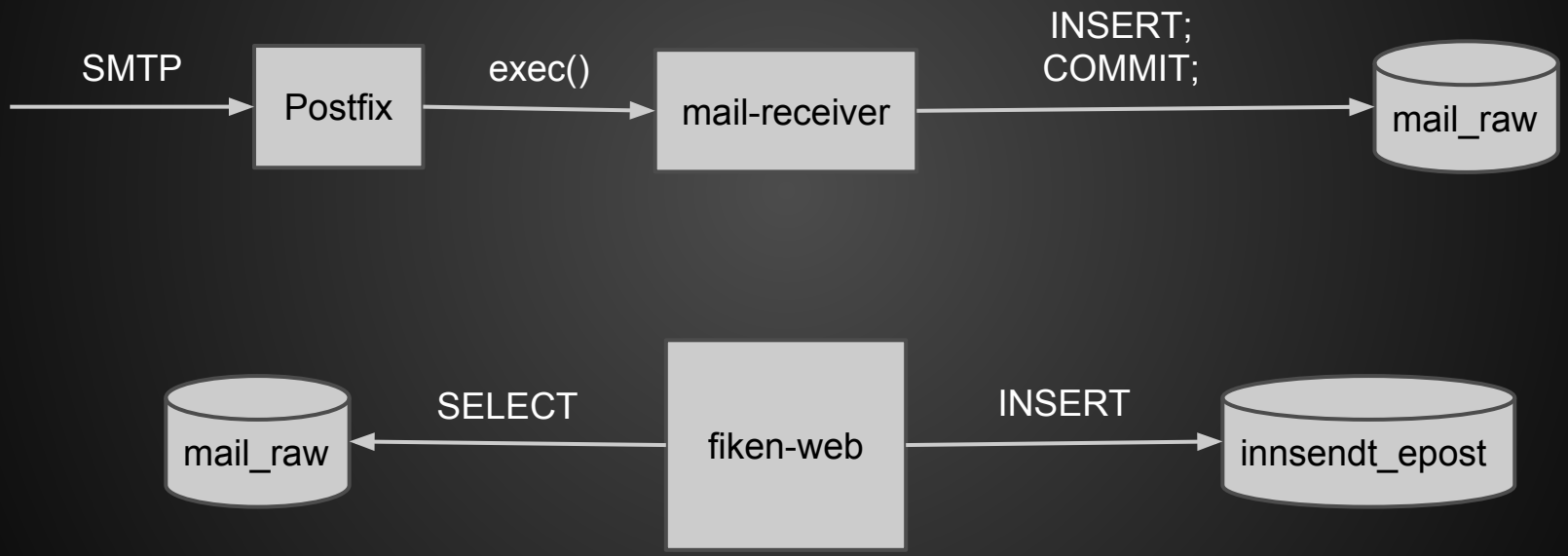
- Trygve Laugstøl
  - trygvis@inamo.no
  - https://twitter.com/trygvis
- Work
  - https://fiken.no
  - https://twitter.com/fiken_no
- Talk
  - Code: github.com/trygvis/javazone-2014
  - Video: https://vimeo.com/105751259
  - Slides: PostgreSQL som MQ - Trygve Laugstøl.pdf
- JavaZone 2014
  - http://2014.javazone.no/presentation.html?id=d63f7405

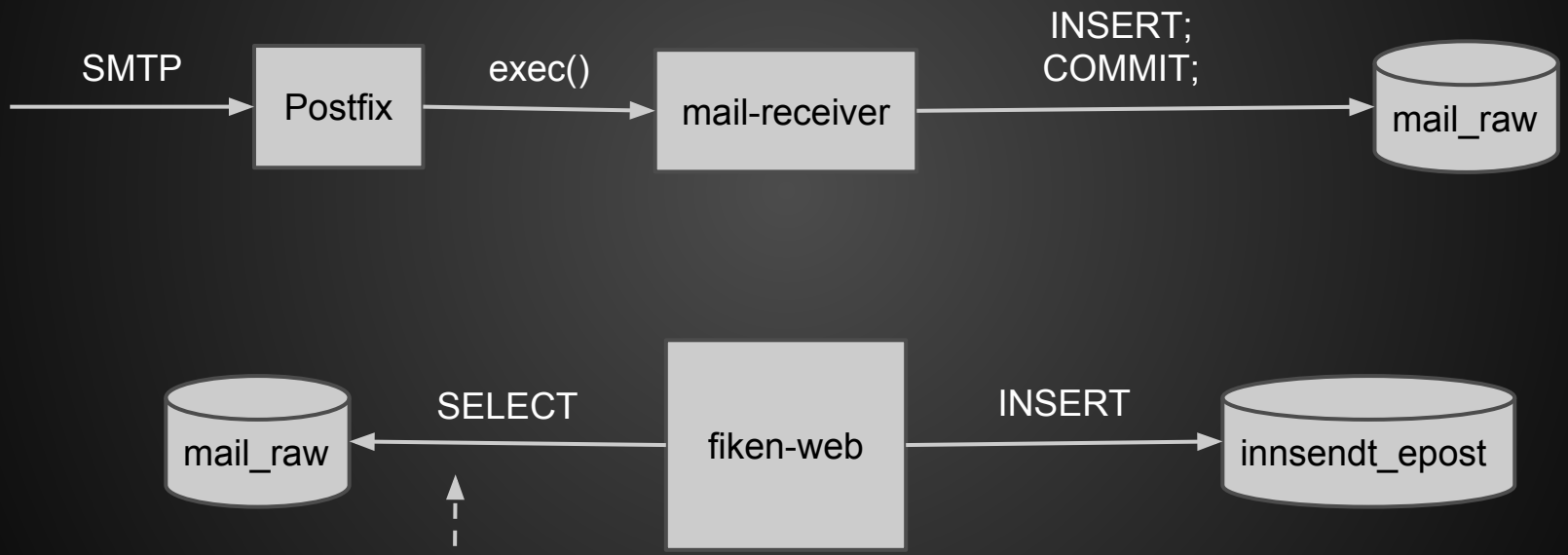
PostgreSQL som MQ

a.k.a Epostmottaket til Fiken.no

Epostmottaket til Fiken.no



Epostmottaket til Fiken.no



1s polling => :(

NOTIFY

Name

NOTIFY -- generate a notification

Synopsis

```
NOTIFY channel [ , payload ]
```

Description

The NOTIFY command sends a notification event together with an optional "payload" string to each client application that has previously executed LISTEN *channel* for the specified channel name in the current database. Notifications are visible to all users.

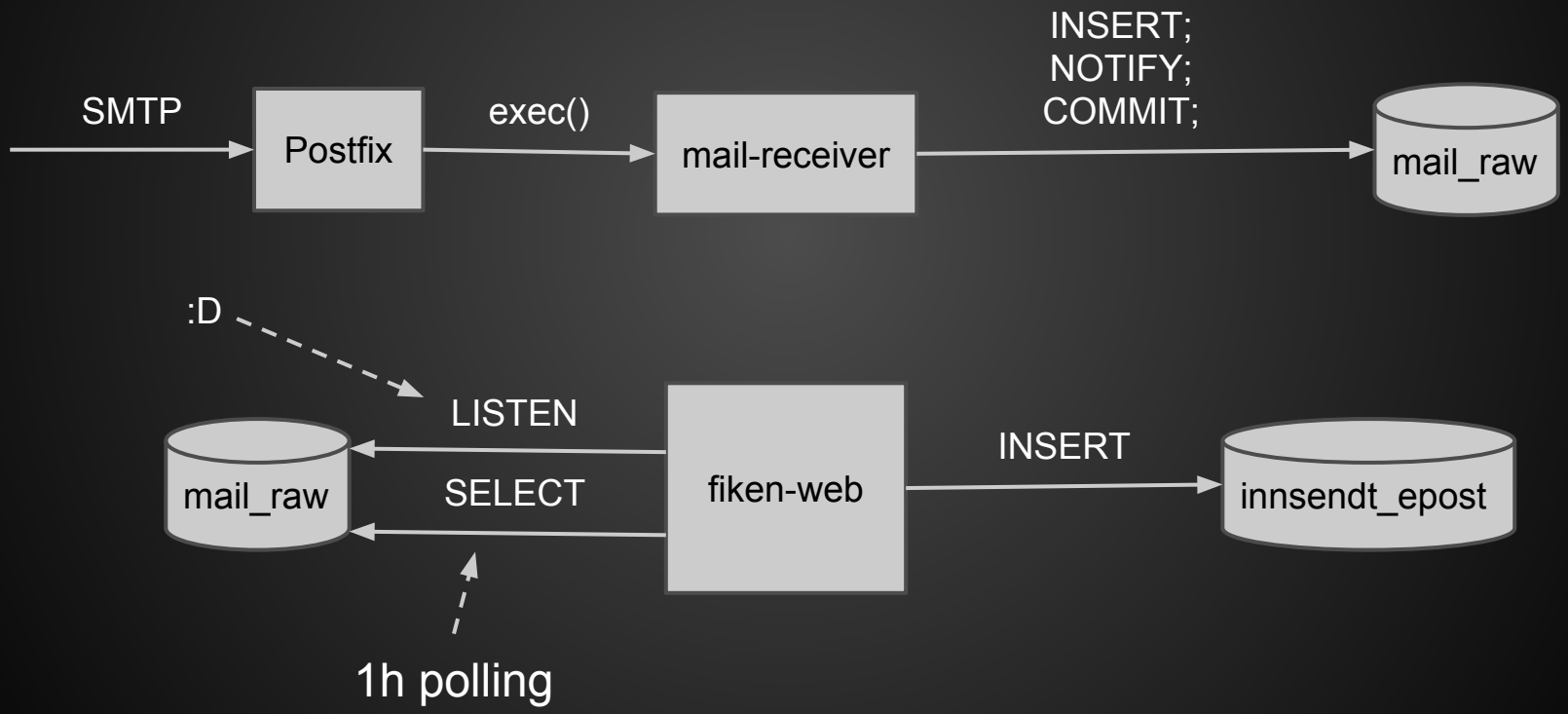
NOTIFY provides a simple interprocess communication mechanism for a collection of processes accessing the same PostgreSQL database. A payload string can be sent along with the notification, and higher-level mechanisms for passing structured data can be built by using tables in the database to pass additional data from notifier to listener(s).

- Mellom backends

- `NOTIFY <channel>[, '<payload>']`
  - Supports transactions
  - payload < 8000 bytes
  - `pg_notify(channel, payload);`

- `LISTEN <channel>`
  - Outside transactions

Epostmottaket til Fiken.no



Chapter 9. PostgreSQL™ Extensions to the JDBC API

Standard LISTEN, NOTIFY, and UNLISTEN commands are issued via the standard Statement interface. To retrieve and process retrieved notifications the Connection must be cast to the PostgreSQL™ specific extension interface PGConnection. From there the getNotifications() method can be used to retrieve any outstanding notifications.

Note

A key limitation of the JDBC driver is that it cannot receive asynchronous notifications and must poll the backend to check if any notifications were issued.

APIer

- PostgreSQL JDBC

```
org.postgresql.PGConnection {
  PGNotification[] getNotifications()
}
```

- pgjdbc-ng

```
com.impossibl.postgres.api.jdbc.PGConnection {
  void addNotificationListener(PGNotificationListener listener)
}
```