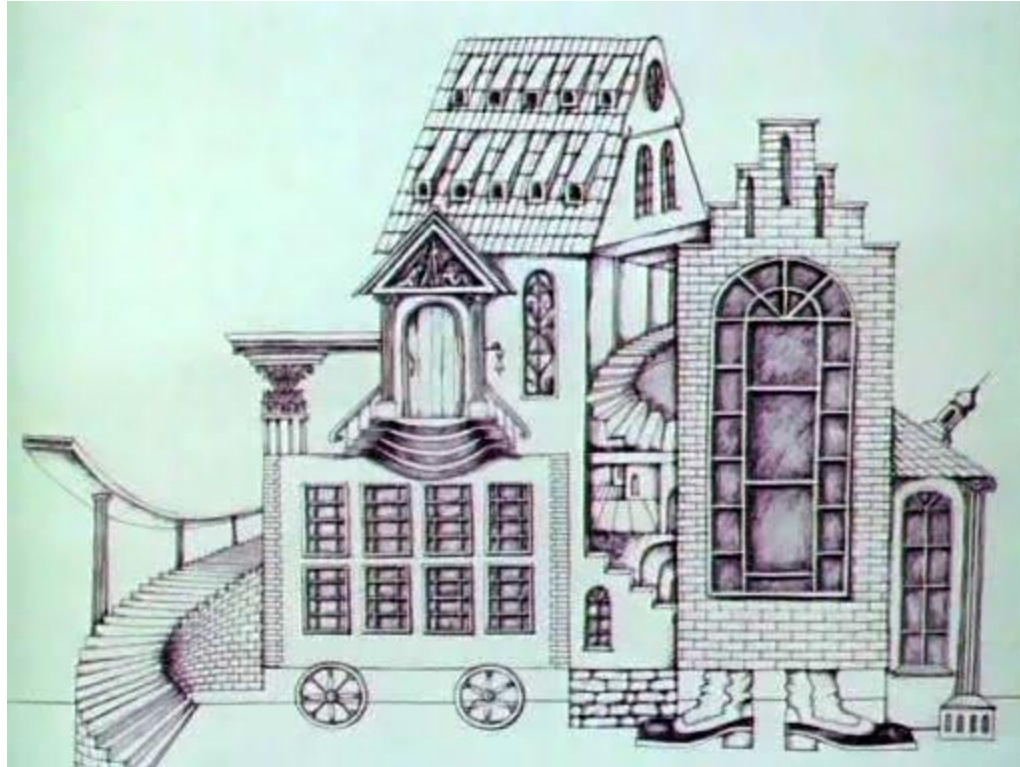


Один из способов быстрого добавления аналитики на python в проект на C++

Александр Боргардт
21.06.2019



199x



20xx-2005



2006-2013



-20xx

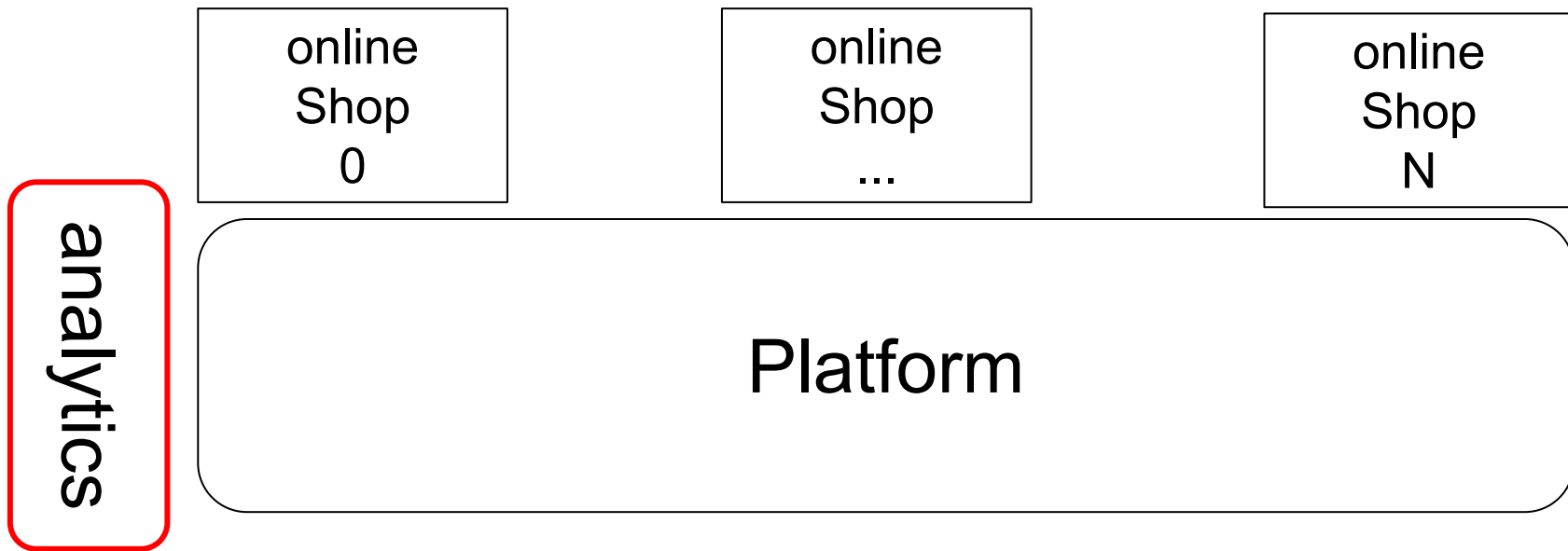


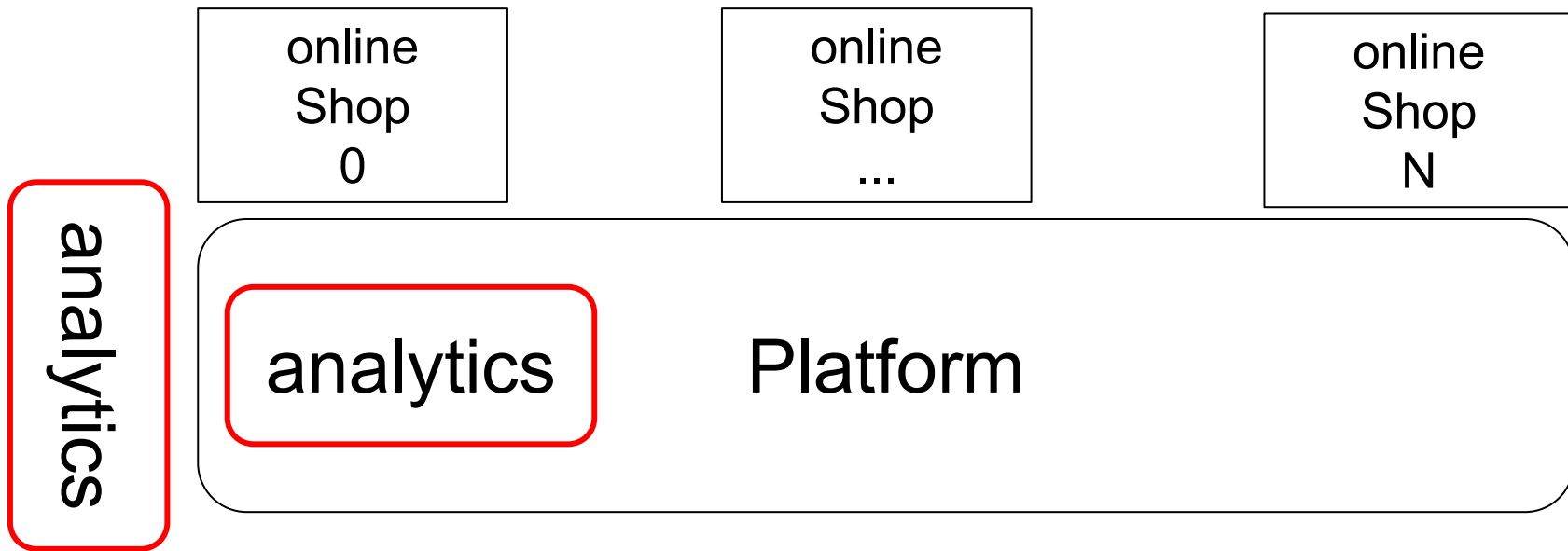
online
Shop
0

online
Shop
...

online
Shop
N







online Shop

online Shop

Back-End

Back-End

Back-End

online Shop

Back-End

Back-End

Back-End

Postgresql Cluster

Примеры метрик:

Примеры метрик:

- Оценка качества закупленного трафика

Примеры метрик:

- Оценка качества закупленного трафика
- Рекомендации для пользователей

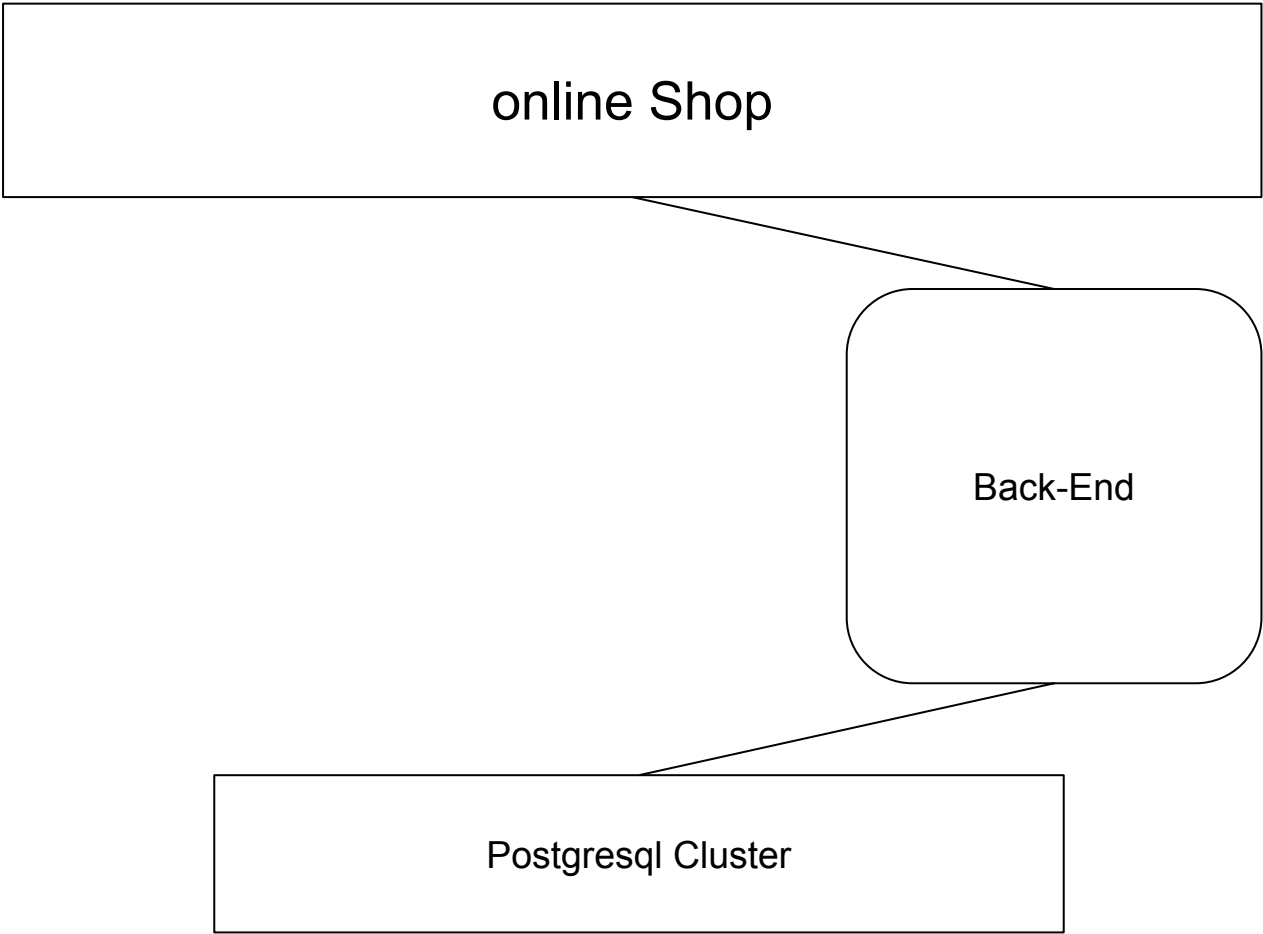
Примеры метрик:

- Оценка качества закупленного трафика
- Рекомендации для пользователей
- Таргетинг

Примеры метрик:

- Оценка качества закупленного трафика
- Рекомендации для пользователей
- Таргетинг
- Поведение пользователя

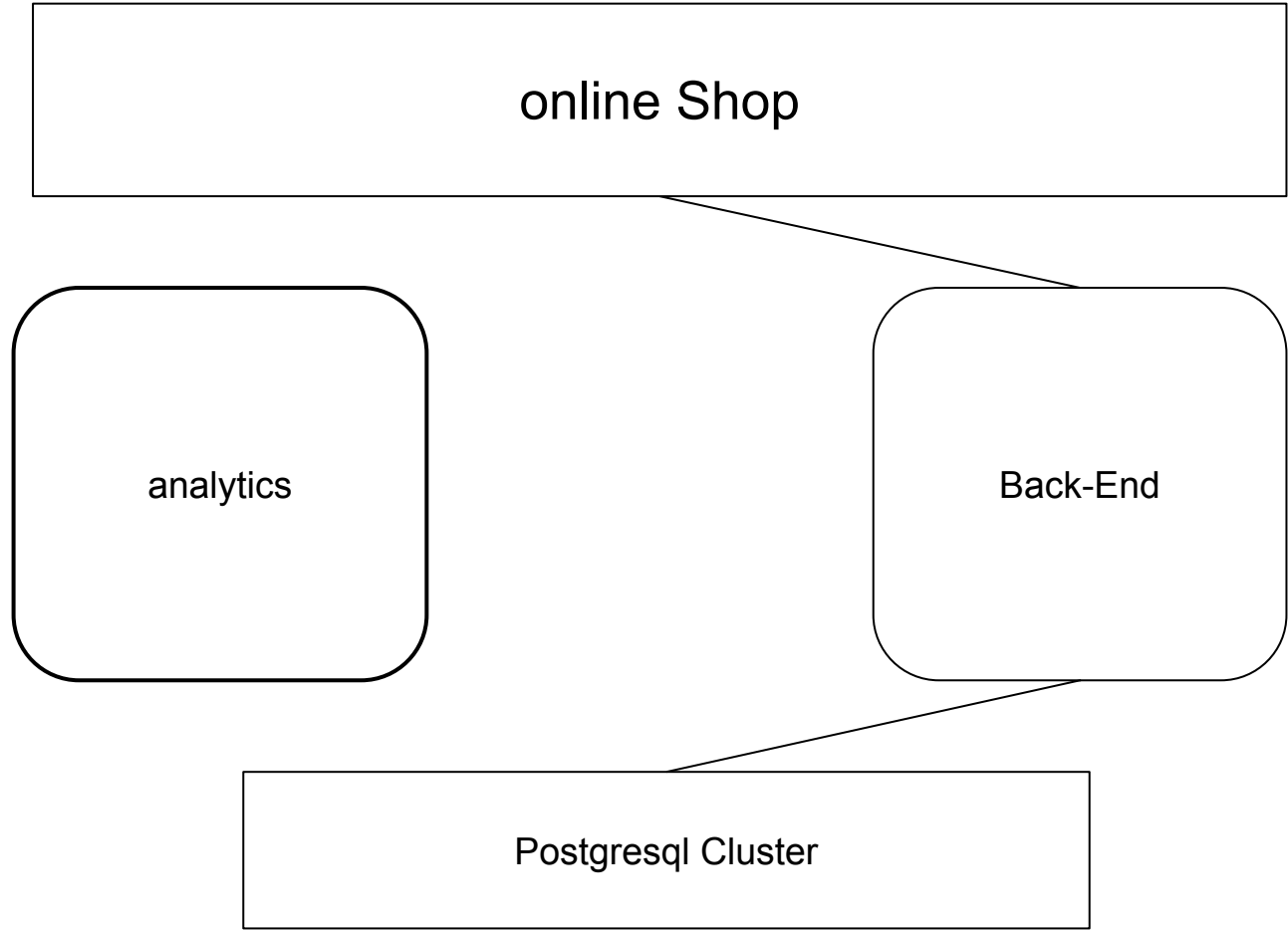
online Shop

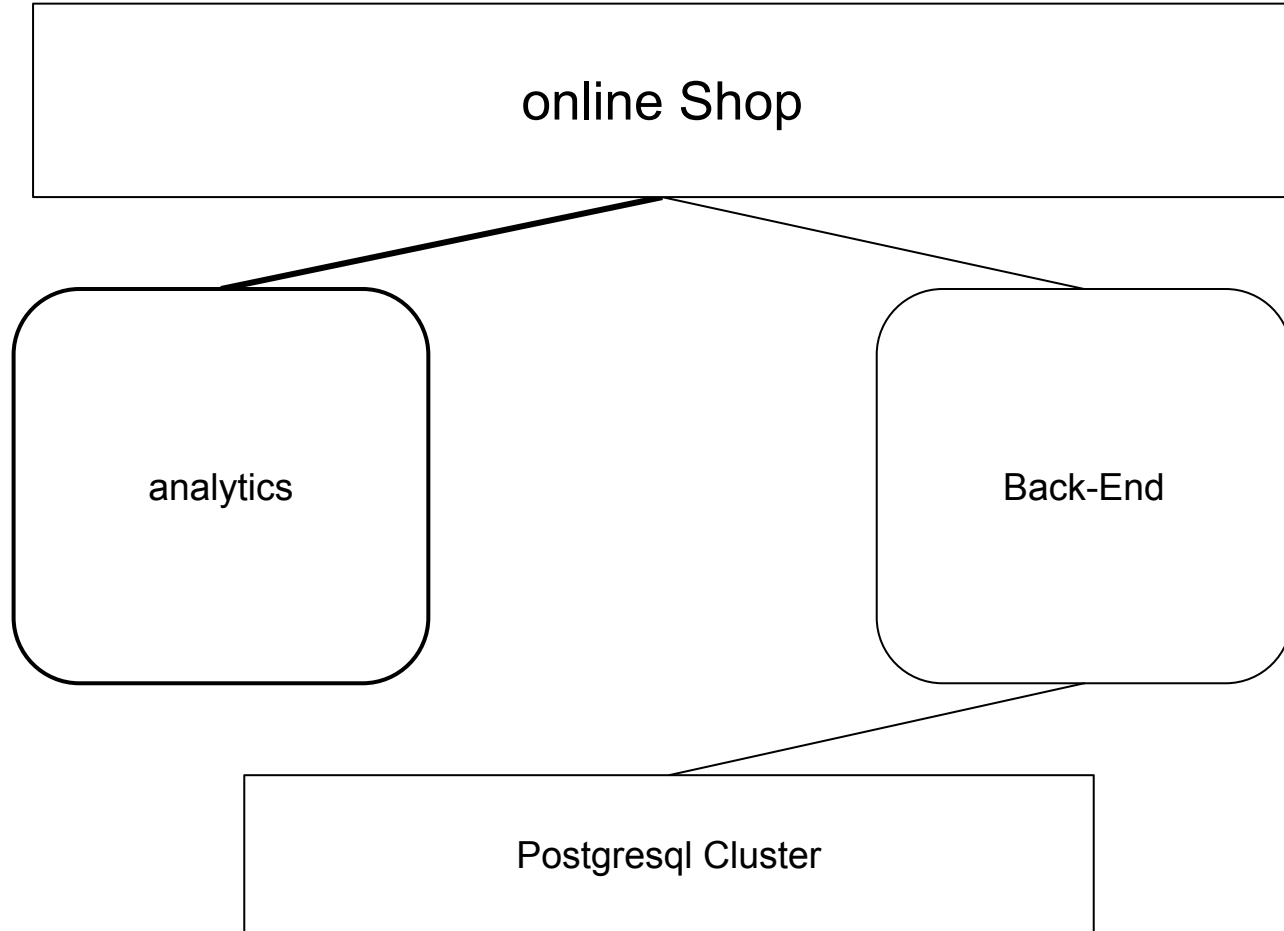


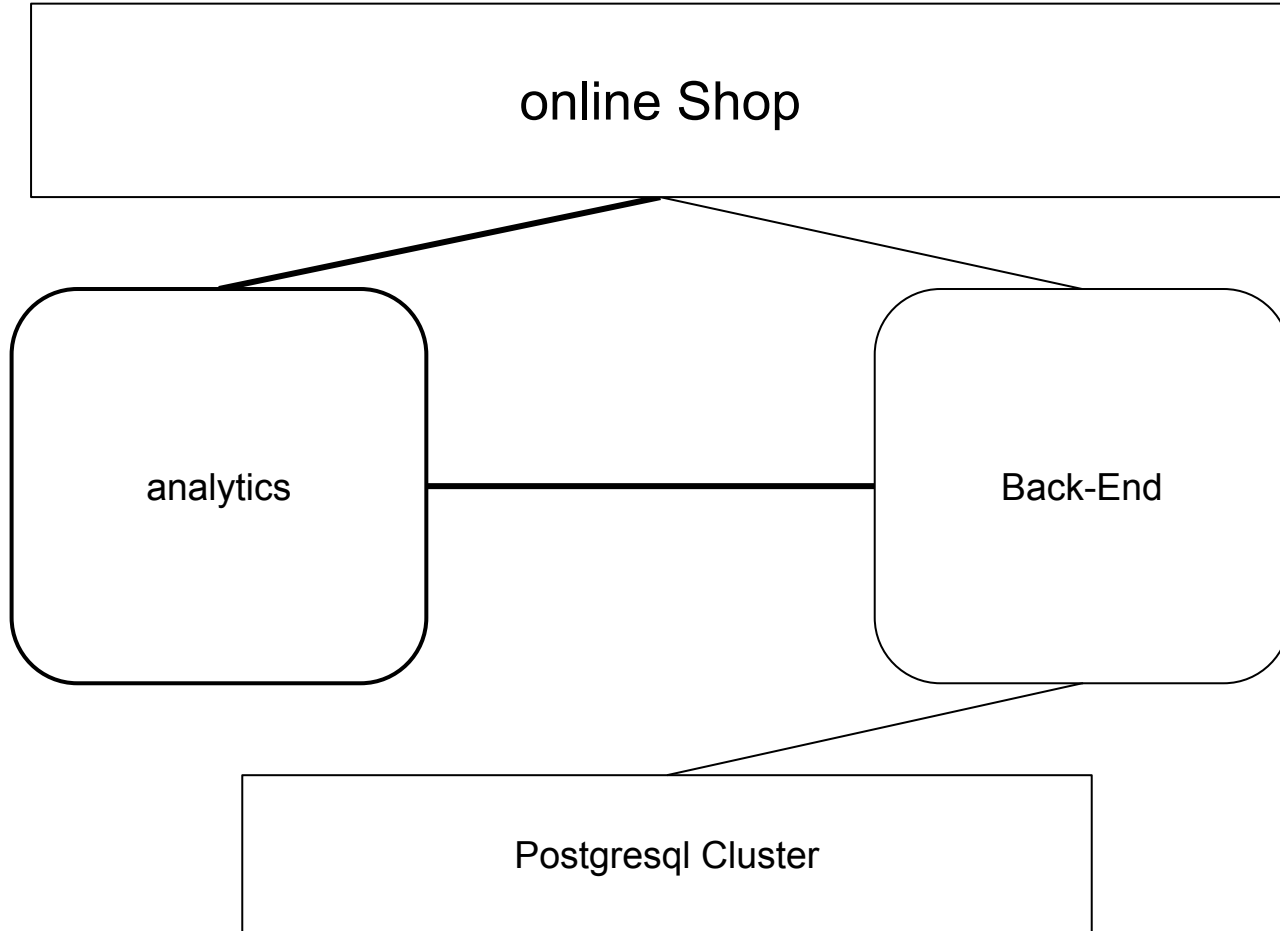
```
graph TD; Shop[online Shop] --- Backend([Back-End]); Backend --- DB[Postgresql Cluster];
```

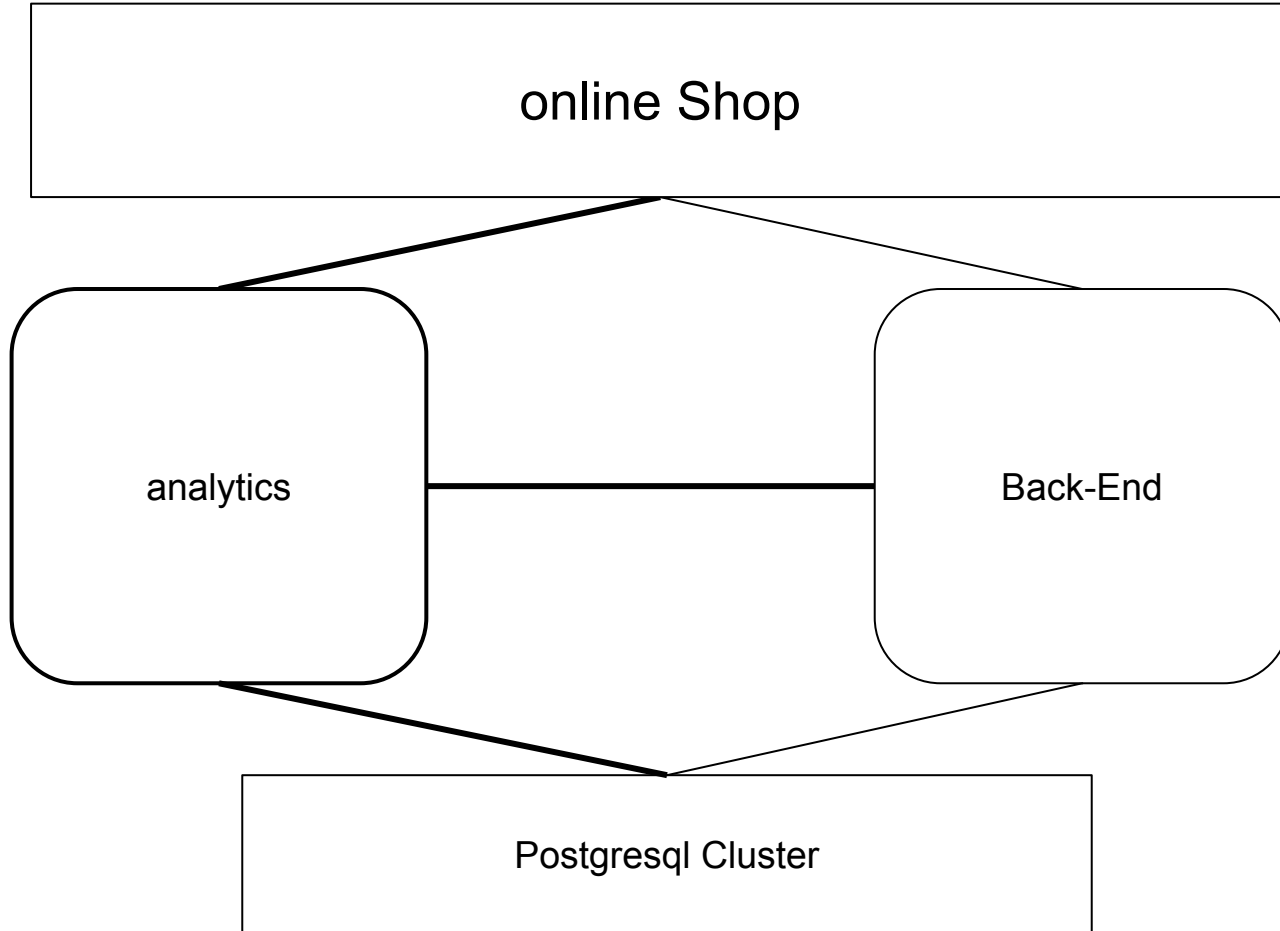
Back-End

Postgresql Cluster









Bad Big Problems:

Bad Big Problems:

- StartUp : 1m - 15m

Bad Big Problems:

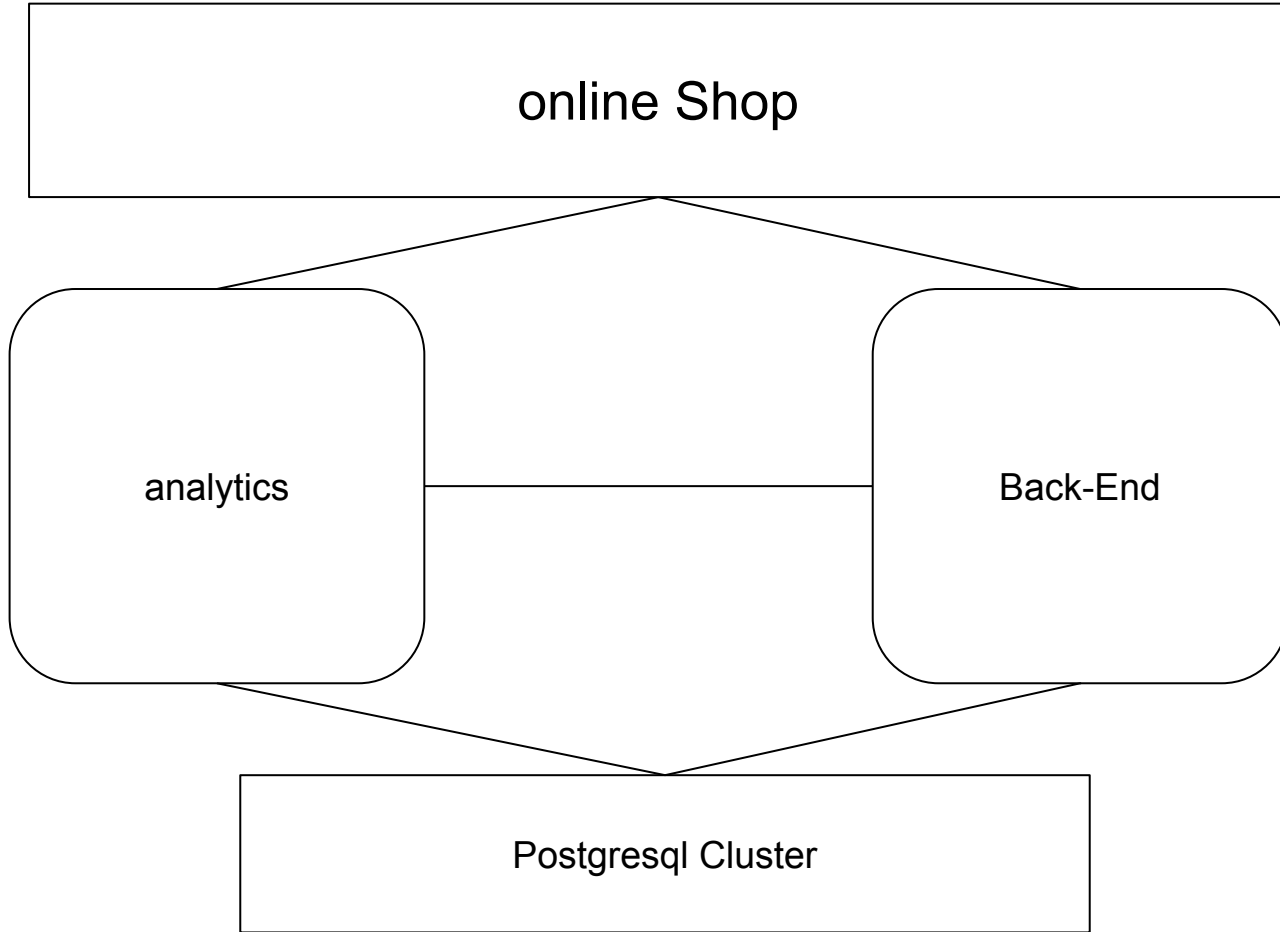
- StartUp : 1m - 15m
- Response latency : 1s - 1m

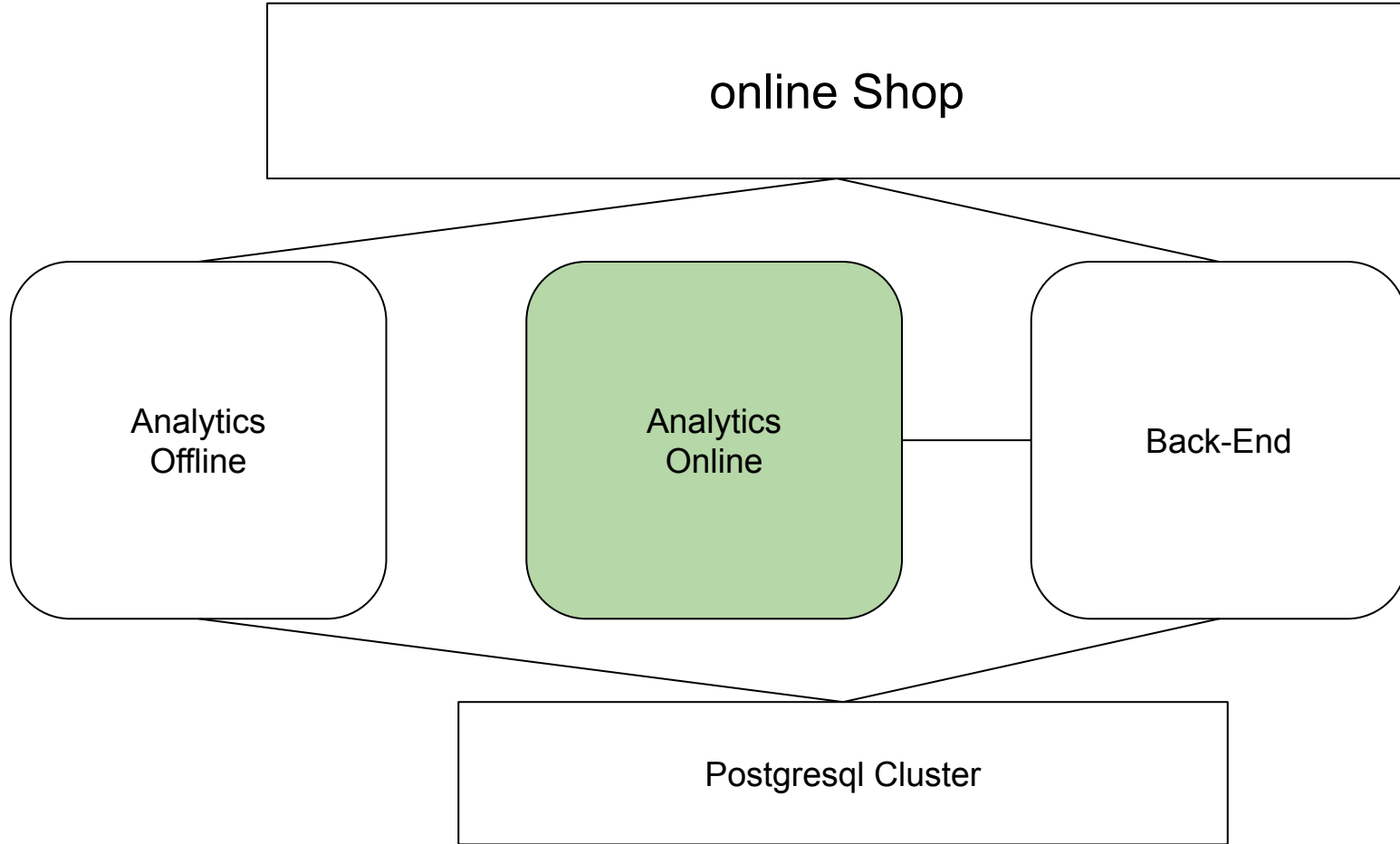
Bad Big Problems:

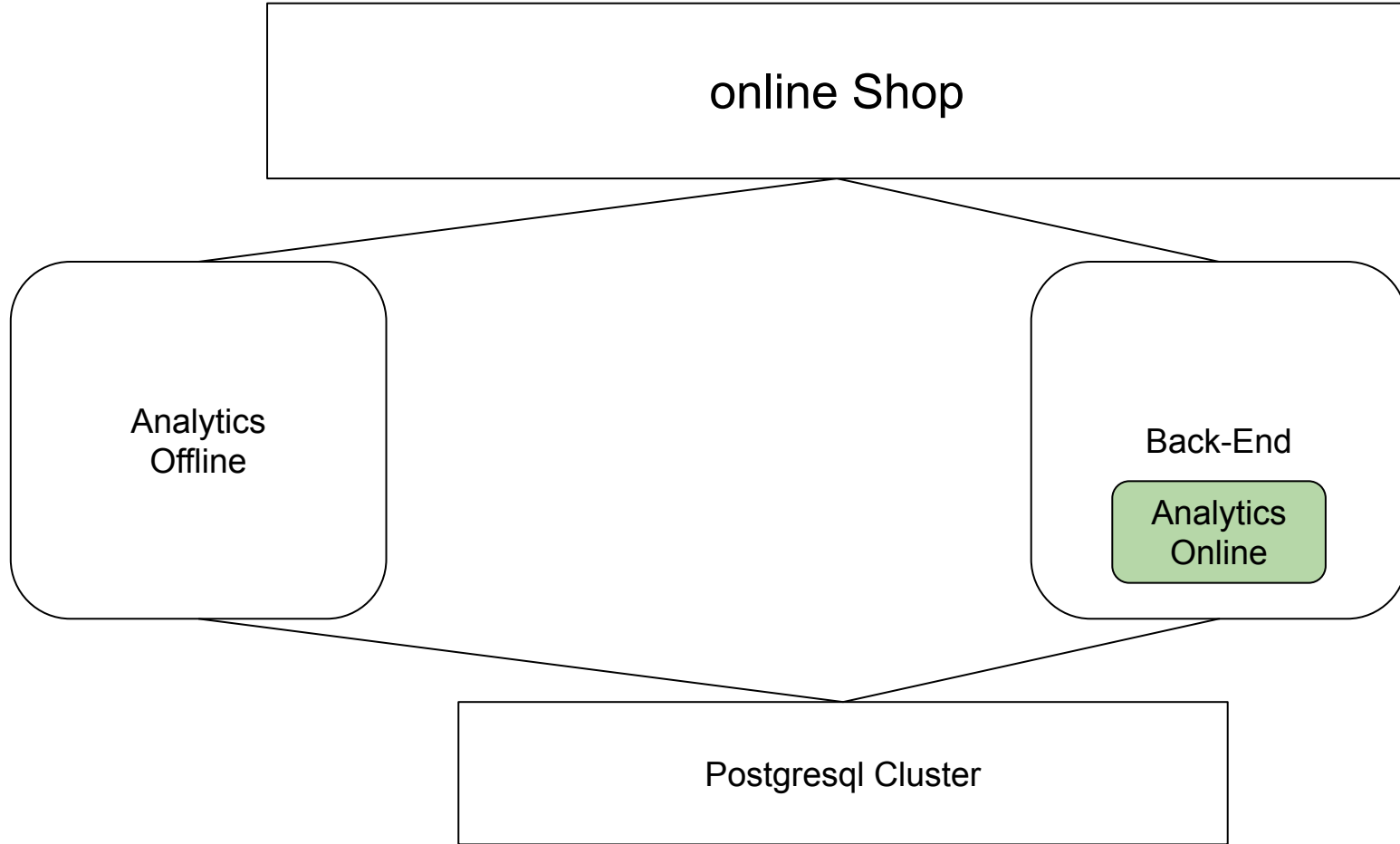
- StartUp : 1m - 15m
- Response latency : 1s - 1m
- ~Python~

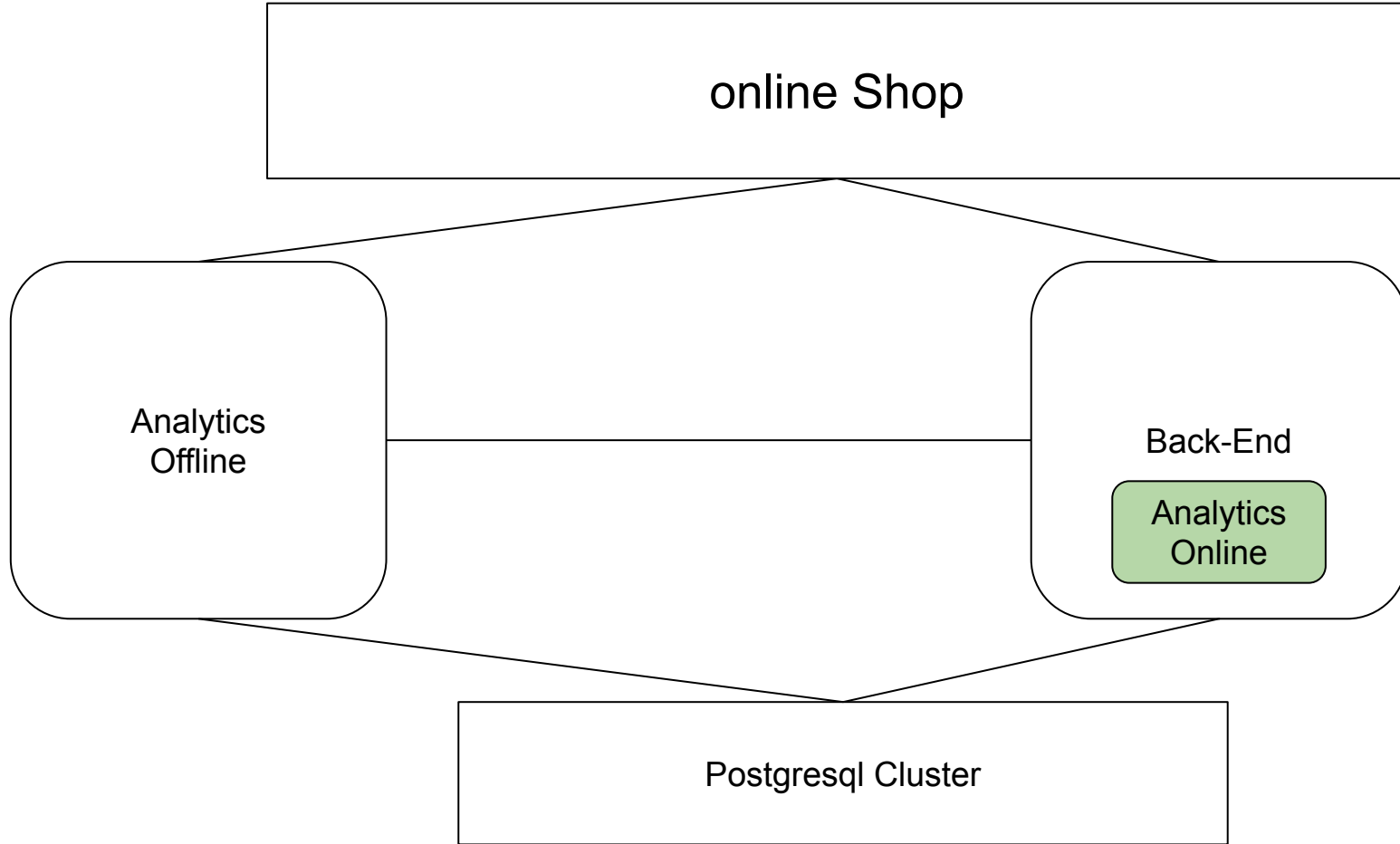
Bad Big Problems:

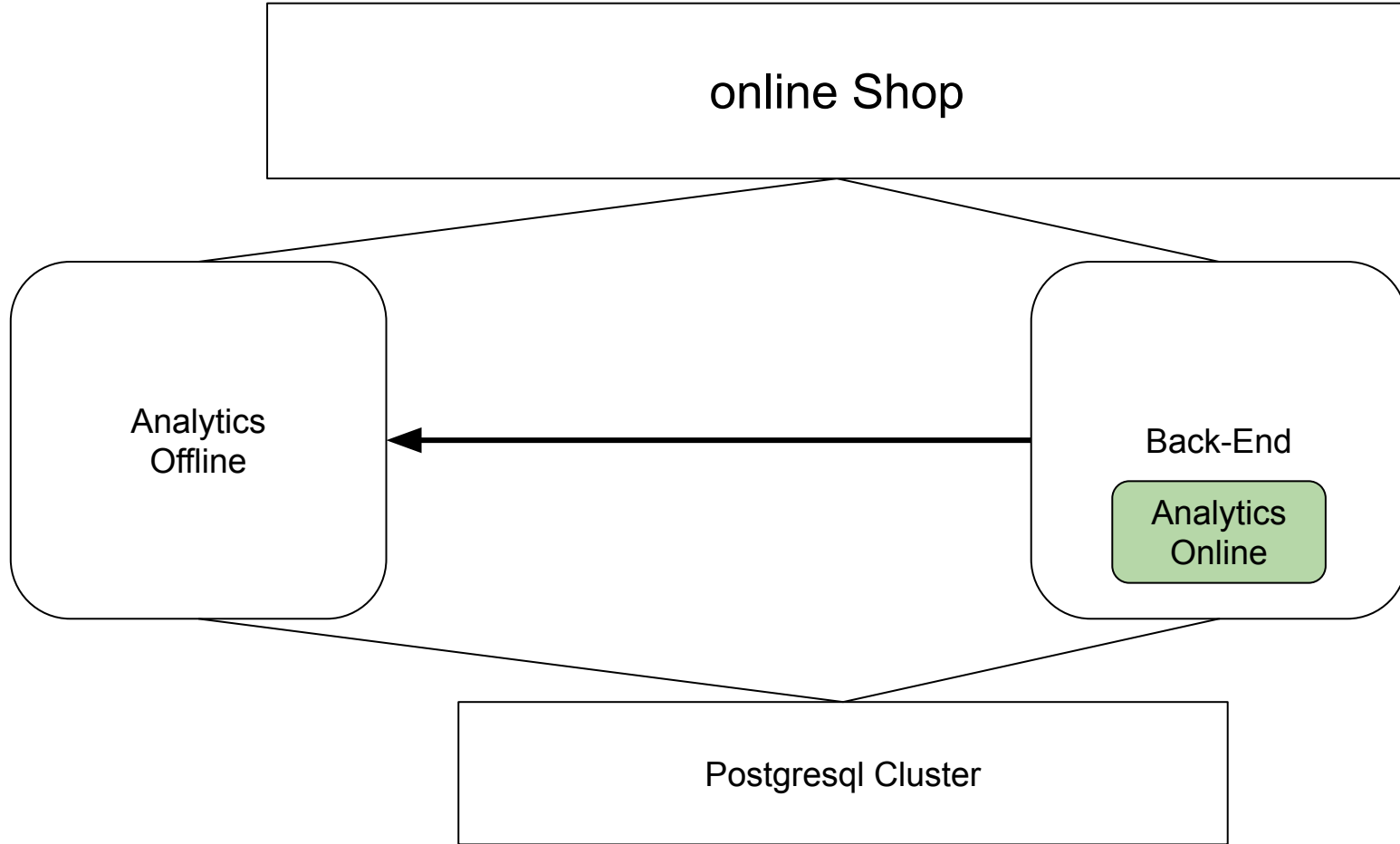
- StartUp : 1m - 15m
- Response latency : 1s - 1m
- ~Python~
- Number of machines: 100











Bad Big Problems:

- StartUp : 1m - 15m
- Response latency : 1s - 1m
- ~Python~
- Number of machines: 100

Bad Big Problems:

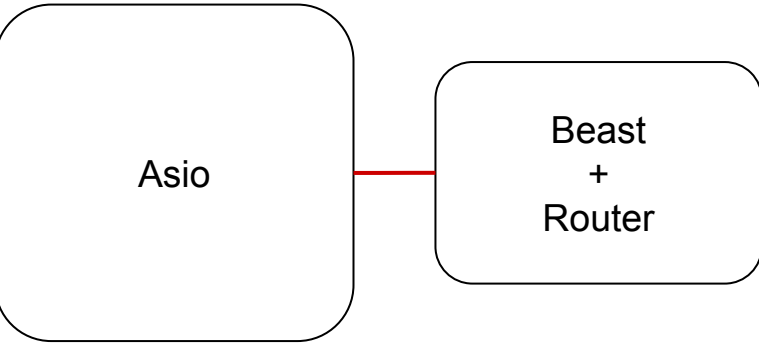
- StartUp : 1m - 15m
- Response latency : 1s - 1m
- ~Python~
- Number of machines: 100

Back-End

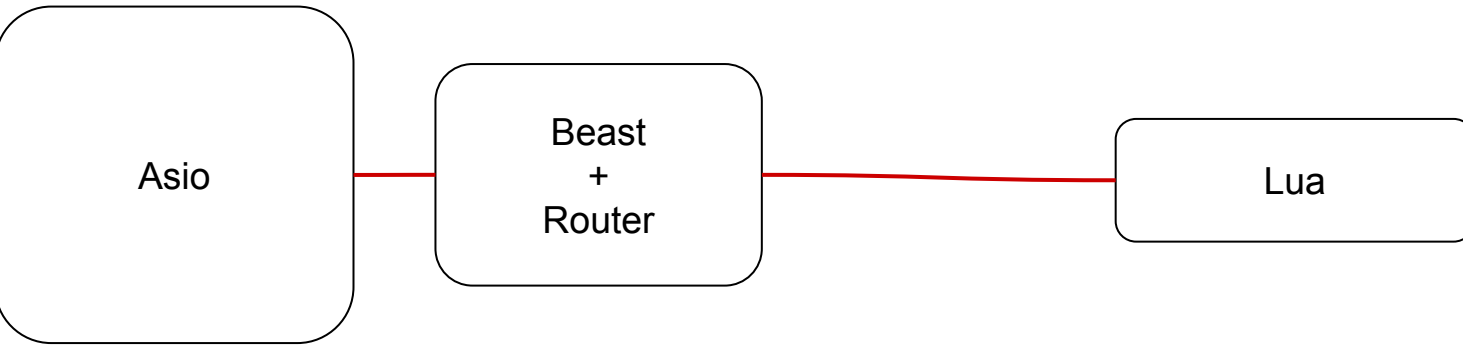
Back-End

Asio

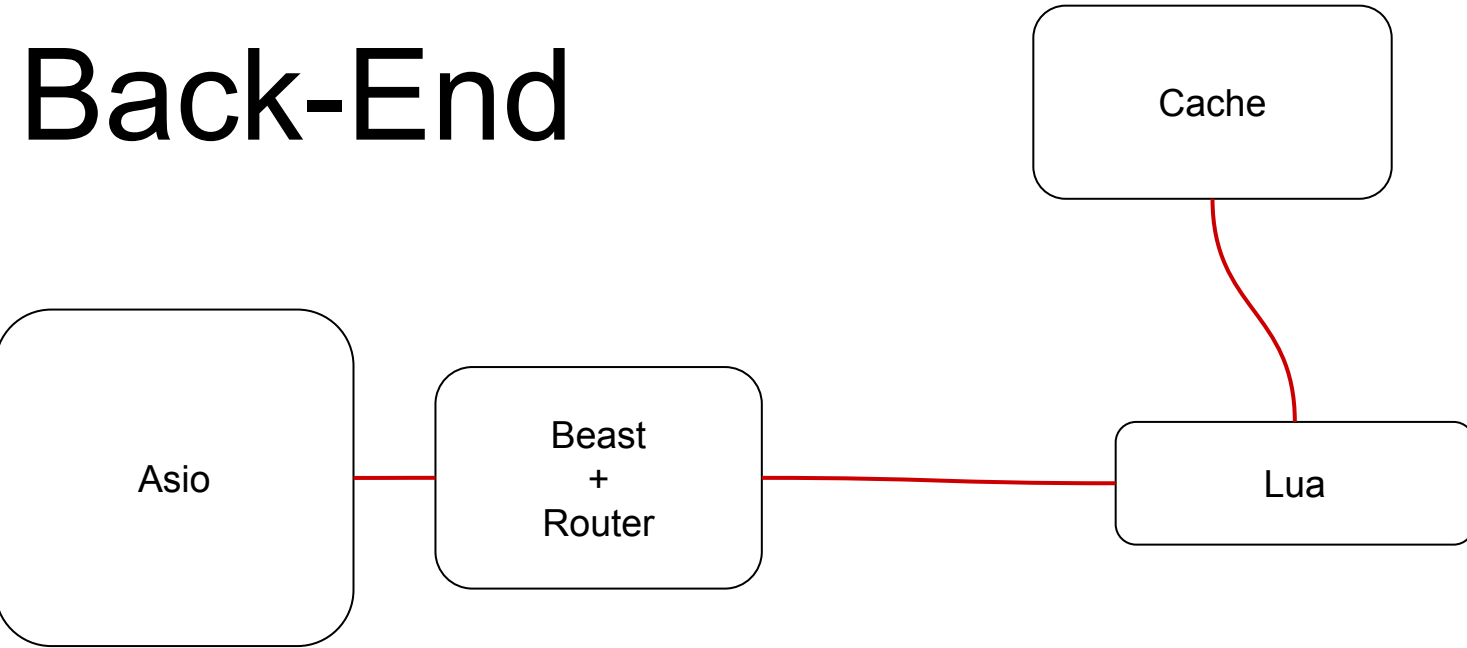
Back-End



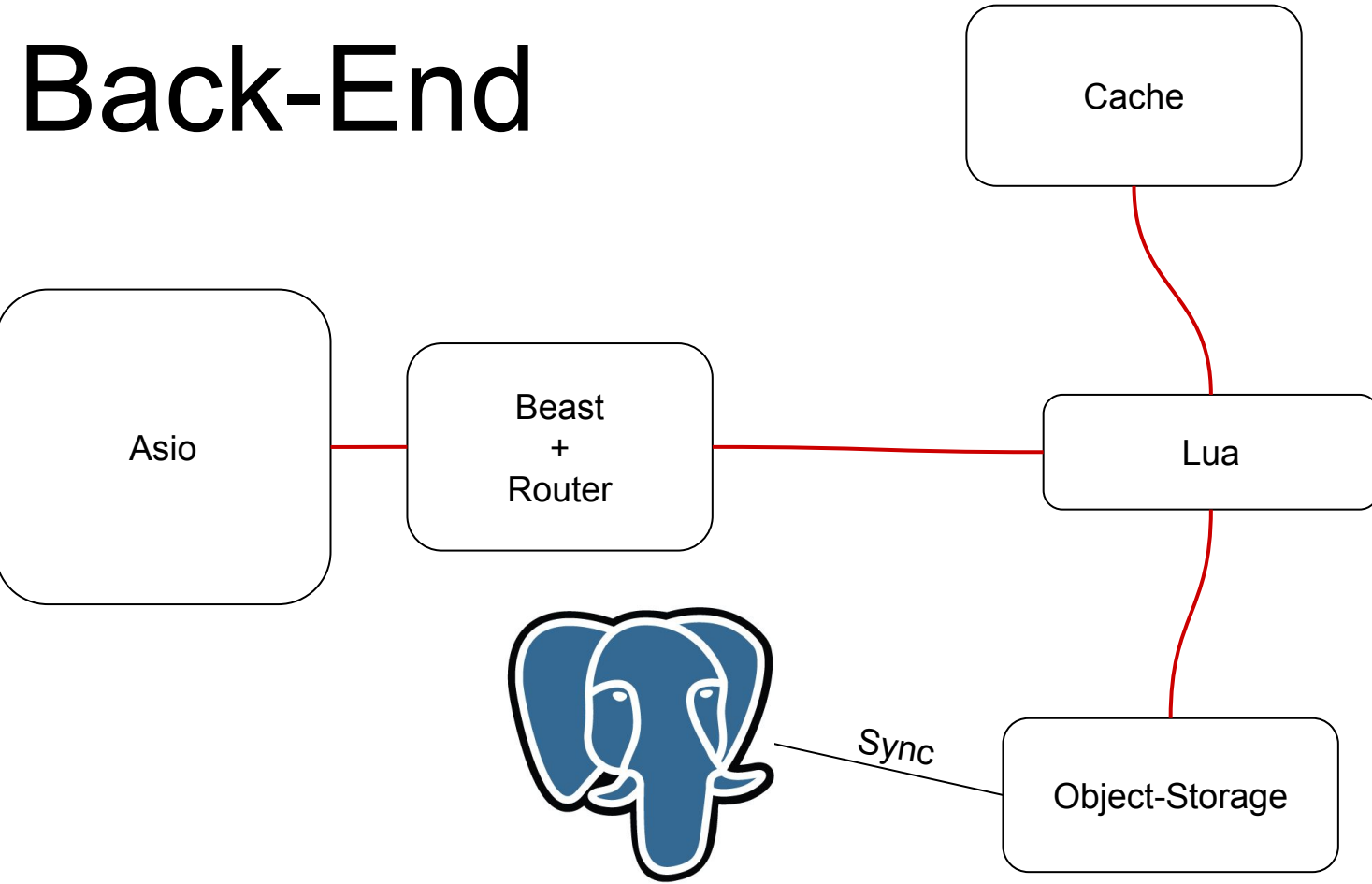
Back-End



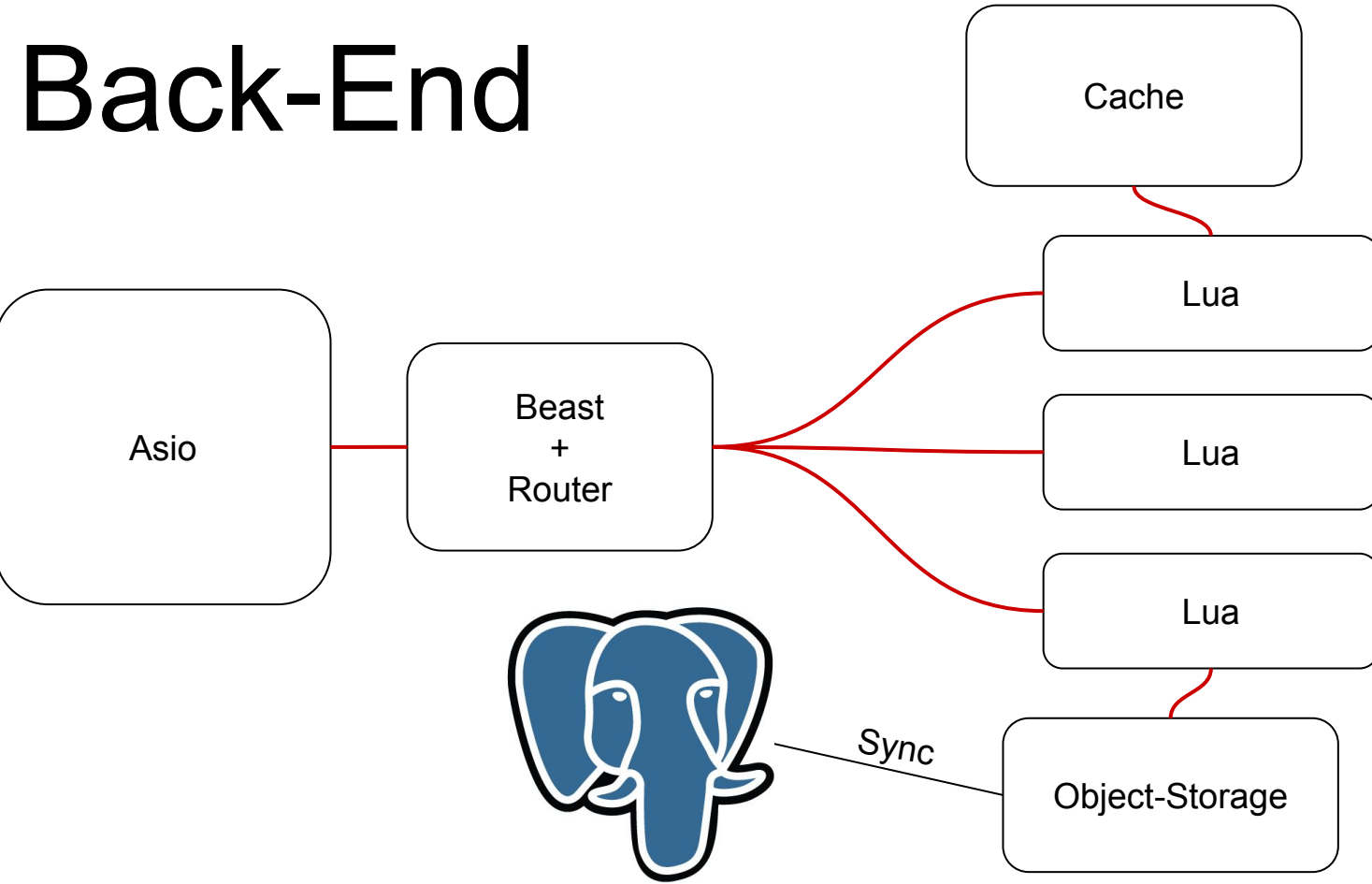
Back-End



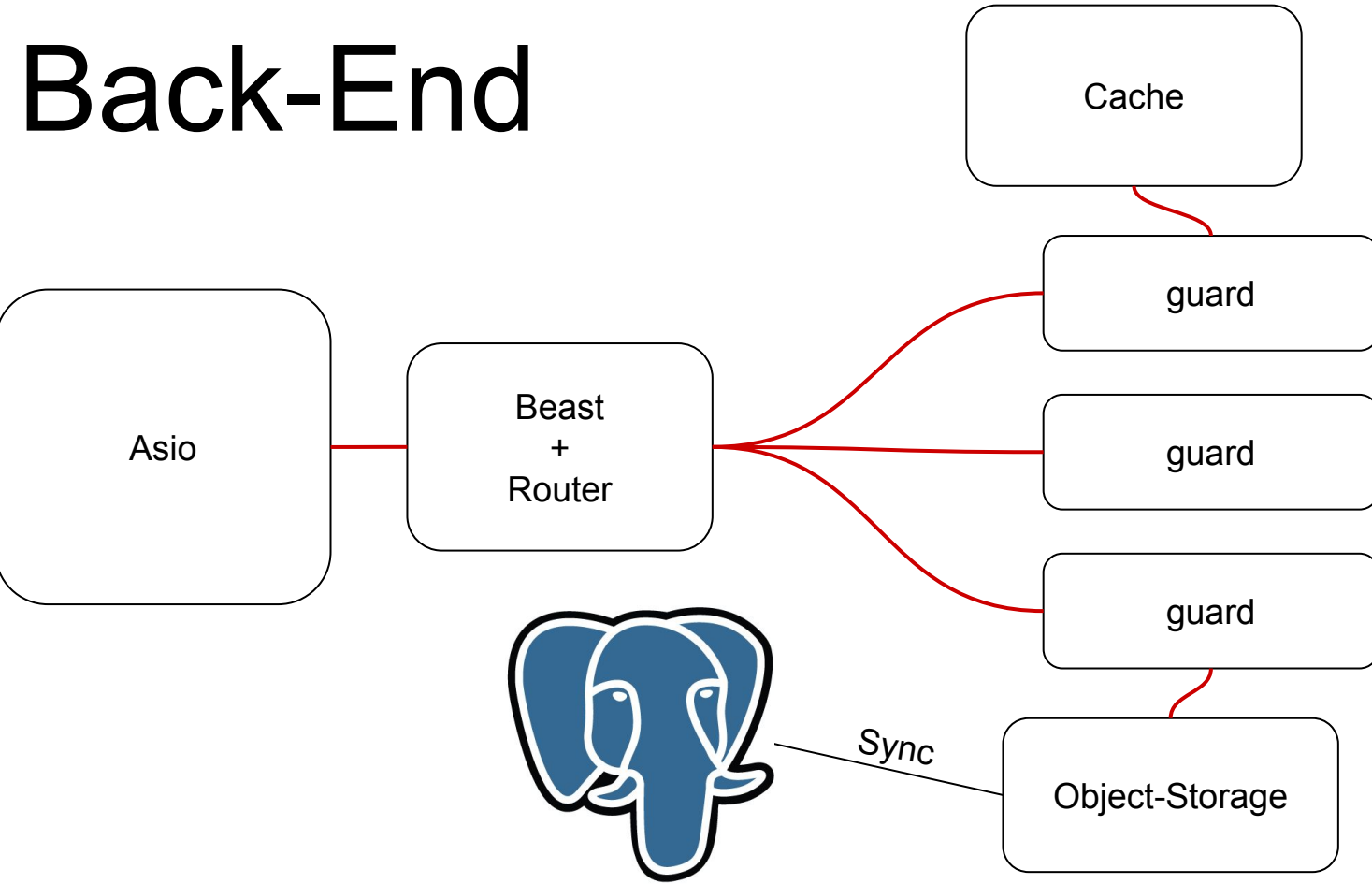
Back-End



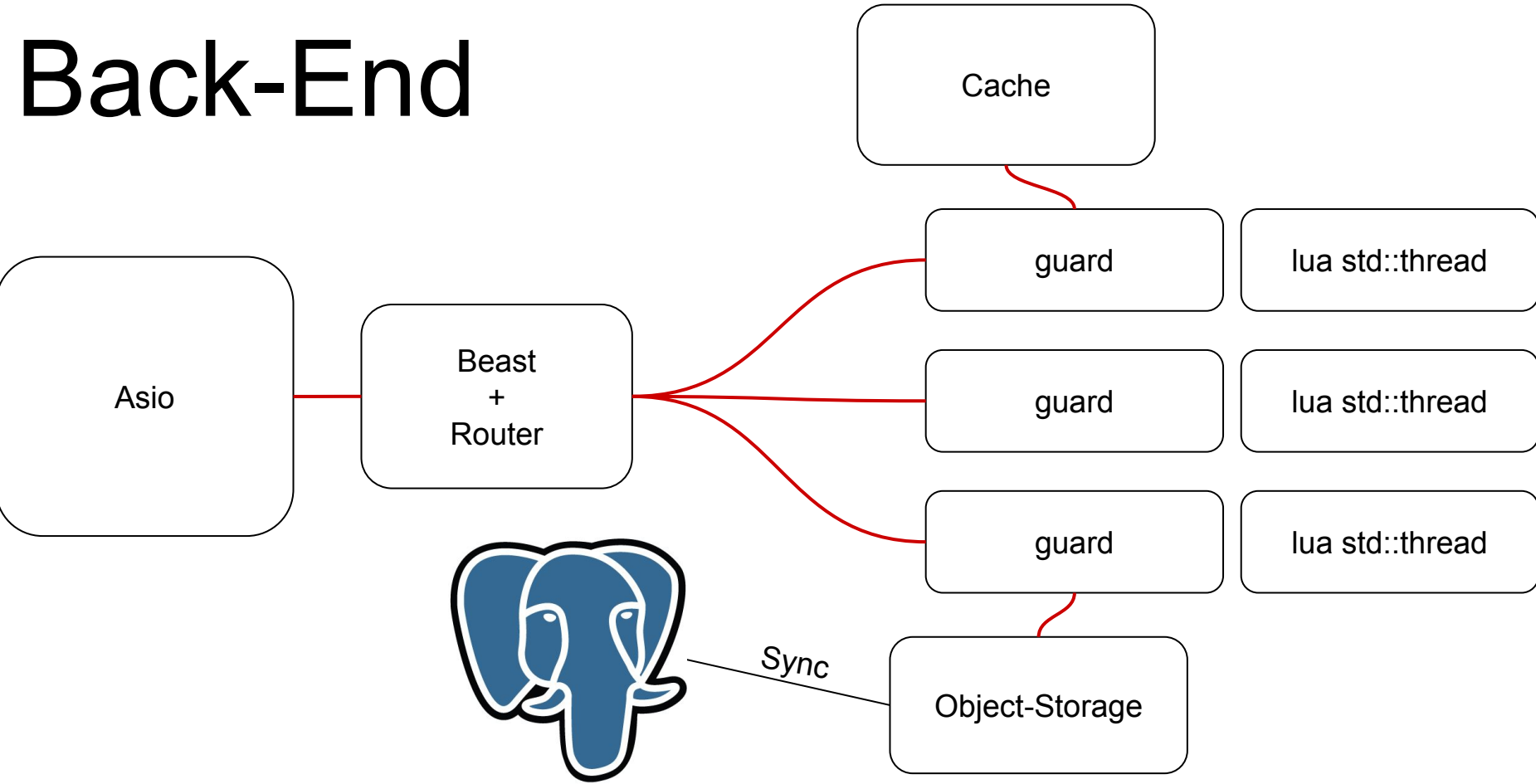
Back-End



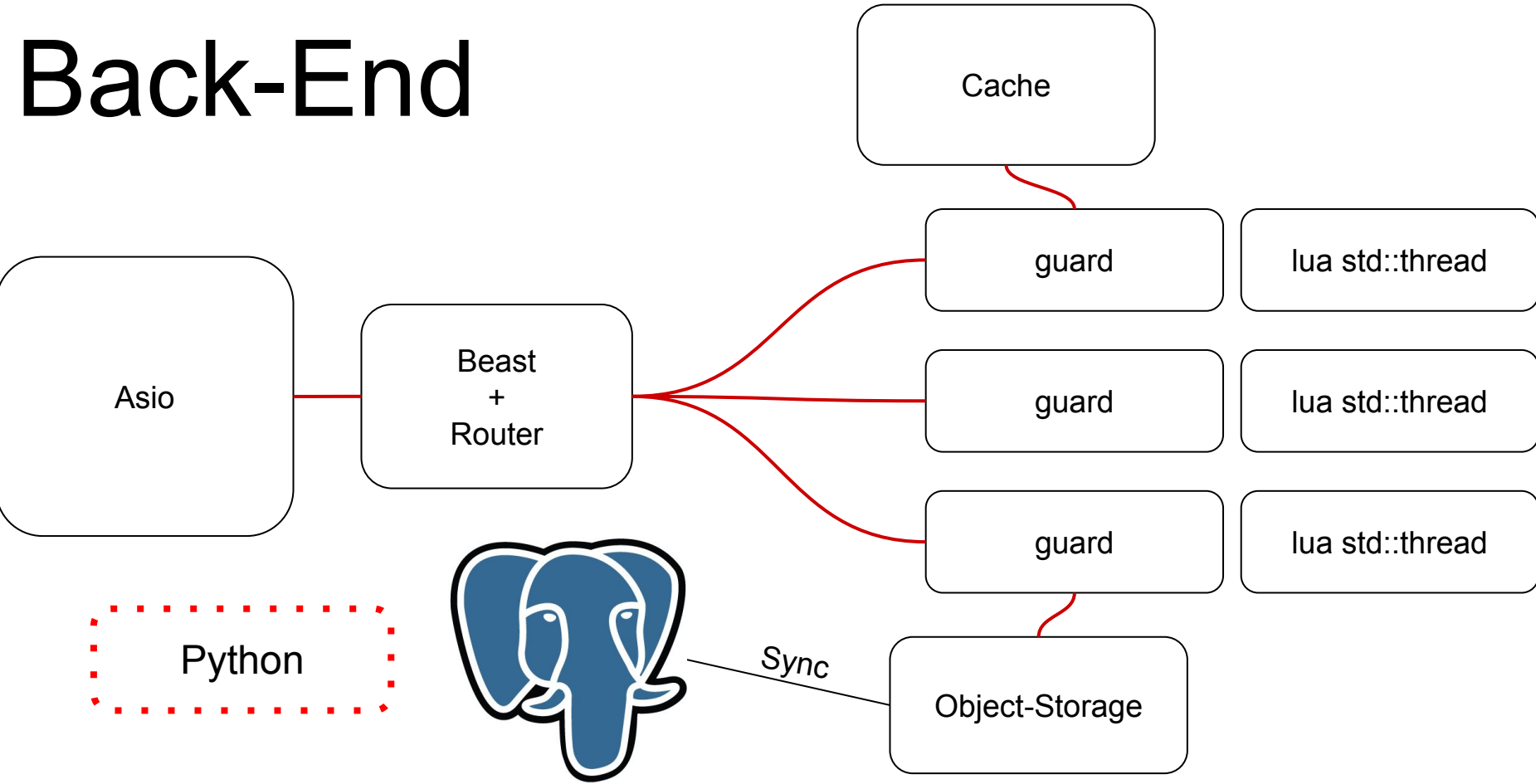
Back-End

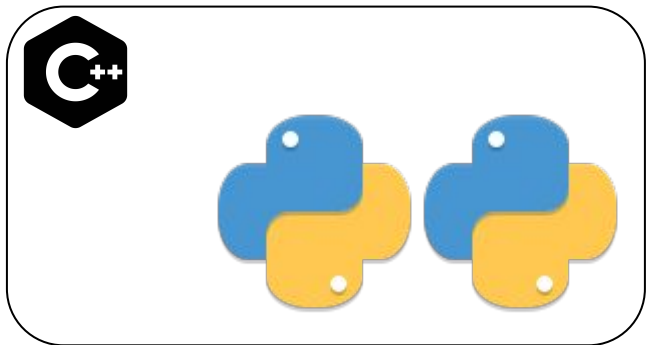


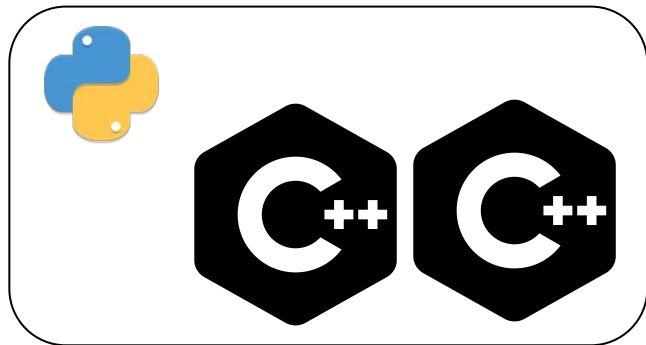
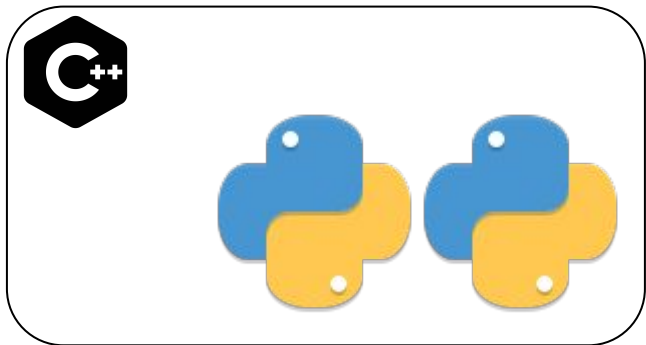
Back-End

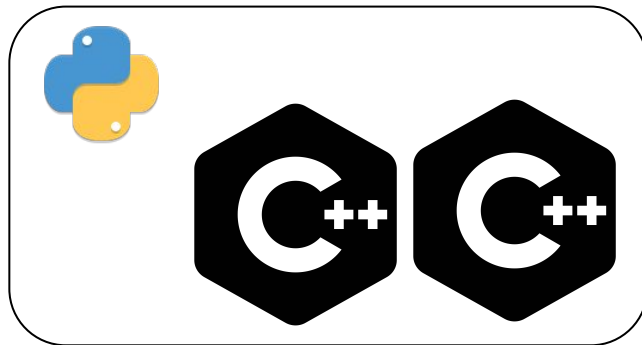
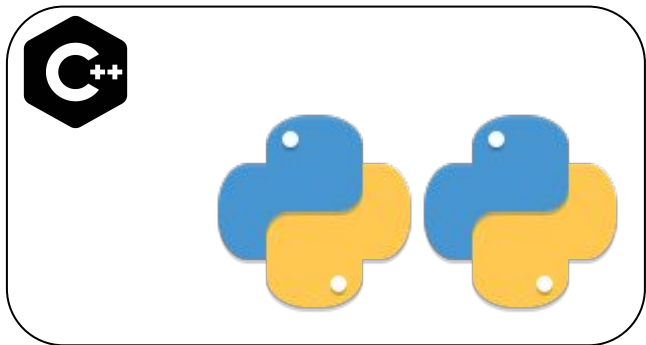


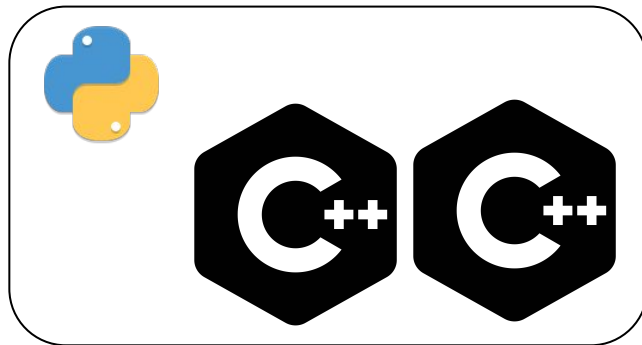
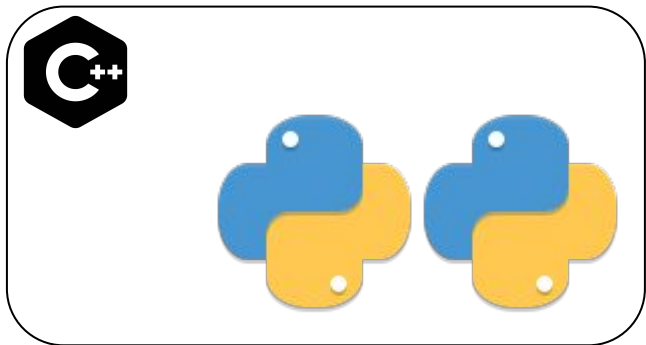
Back-End







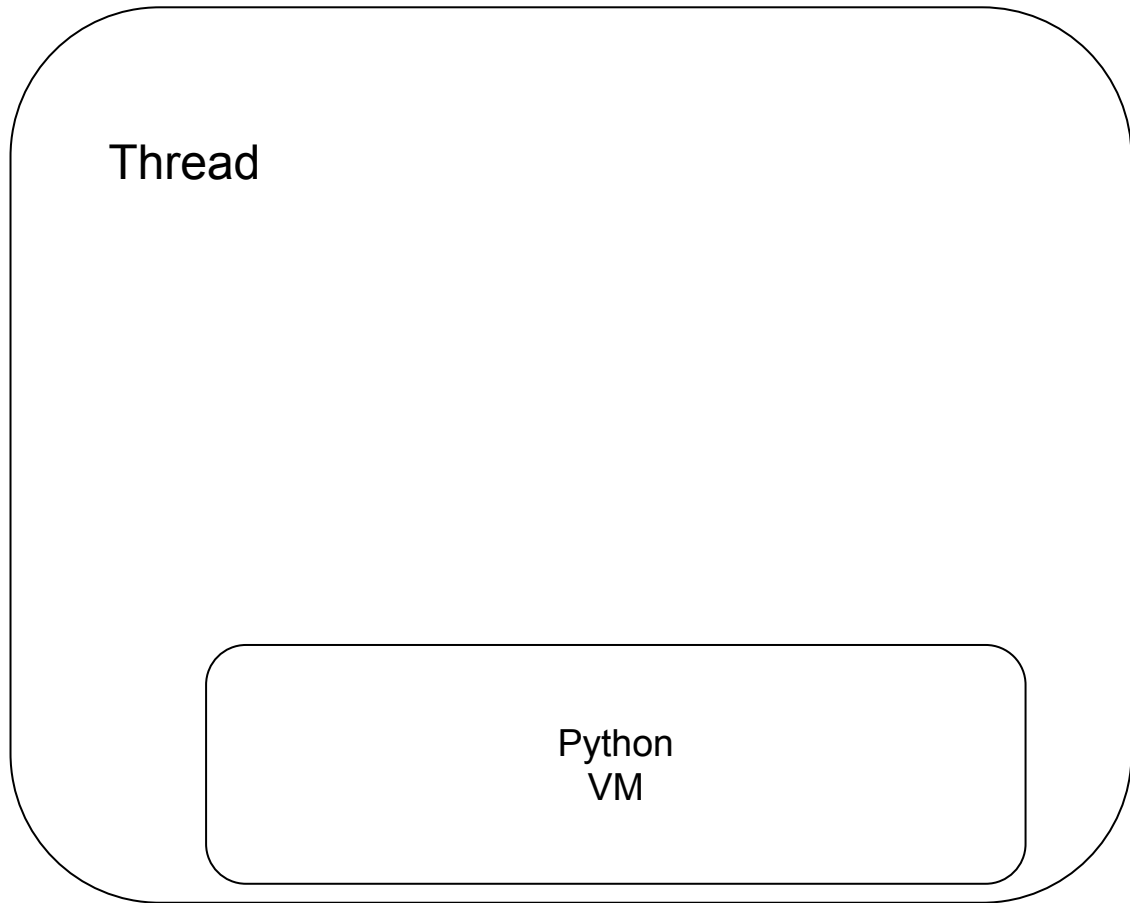




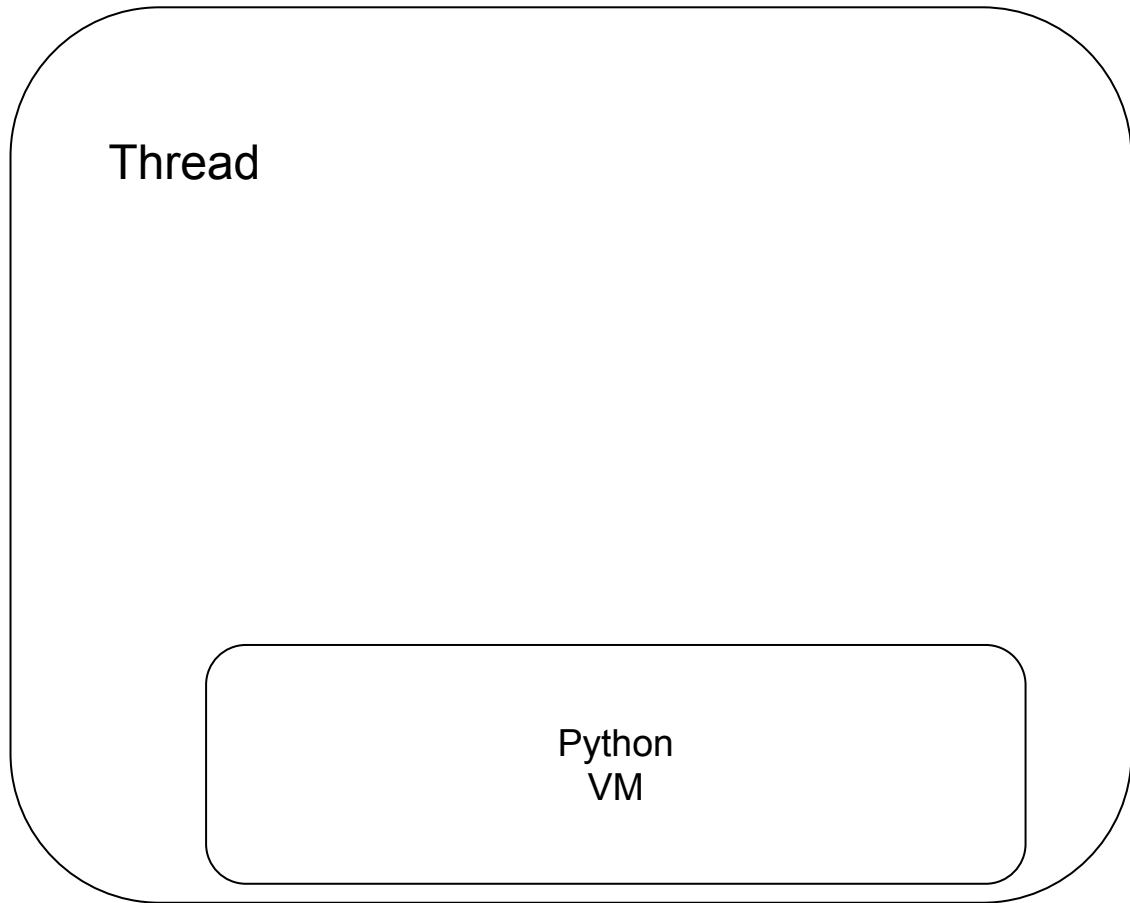
Standalone Thread Python VM

Thread

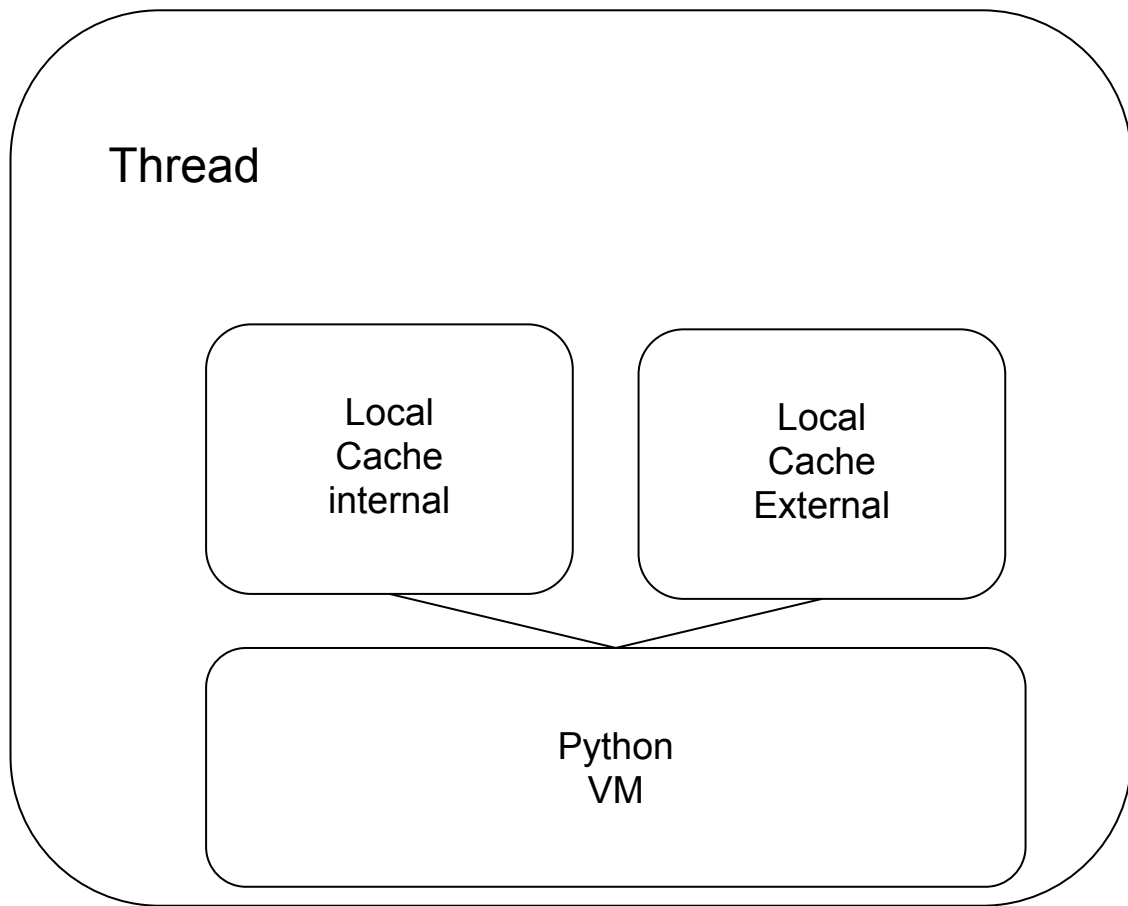
Standalone Thread Python VM



Standalone Thread Python VM



Standalone Thread Python VM



- threads + mutex +.....

- threads + mutex +..... -> difficult path

- threads + mutex +..... -> difficult path
- FSM

- threads + mutex +..... -> difficult path
- FSM -> What to take from the library ?

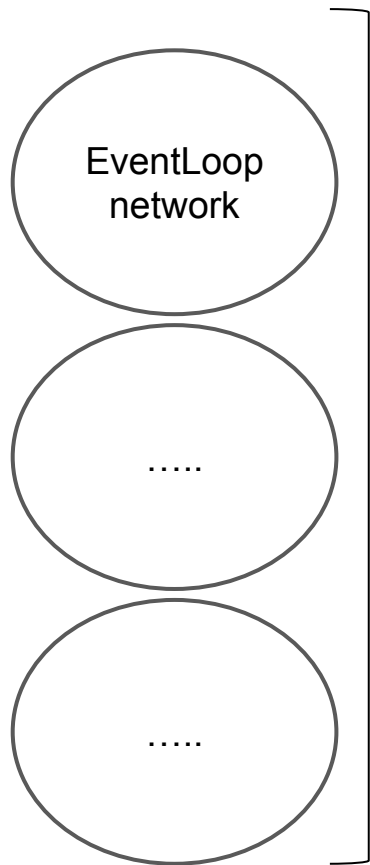
- threads + mutex +..... -> difficult path
- FSM -> What to take from the library ?
- Coroutines

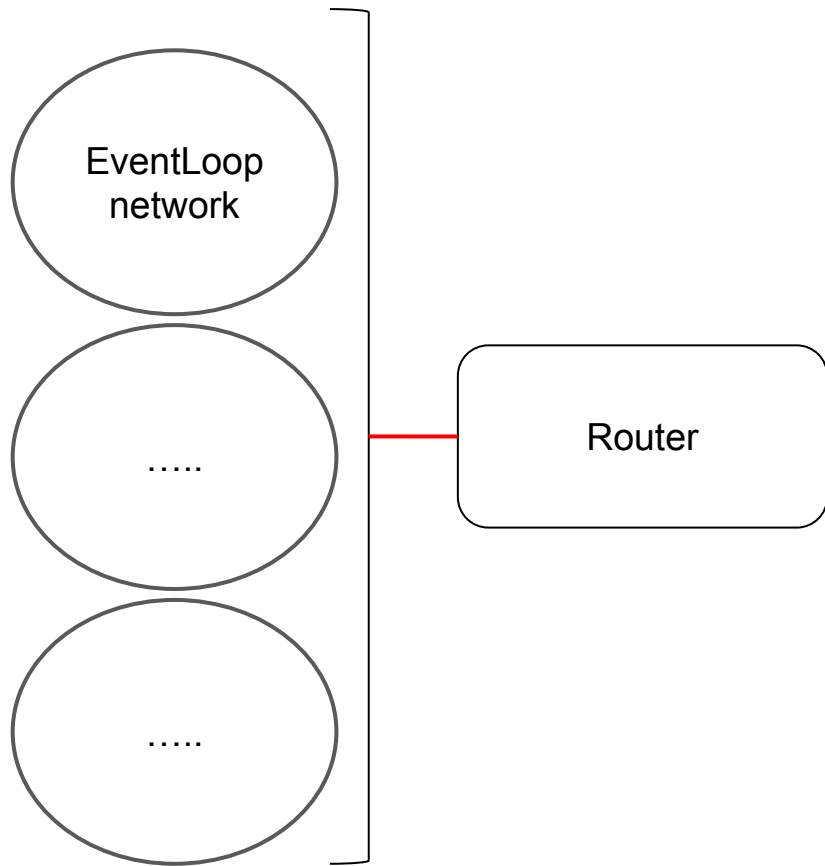
- threads + mutex +..... -> difficult path
- FSM -> What to take from the library ?
- Coroutines -> ? C++ 20 ?

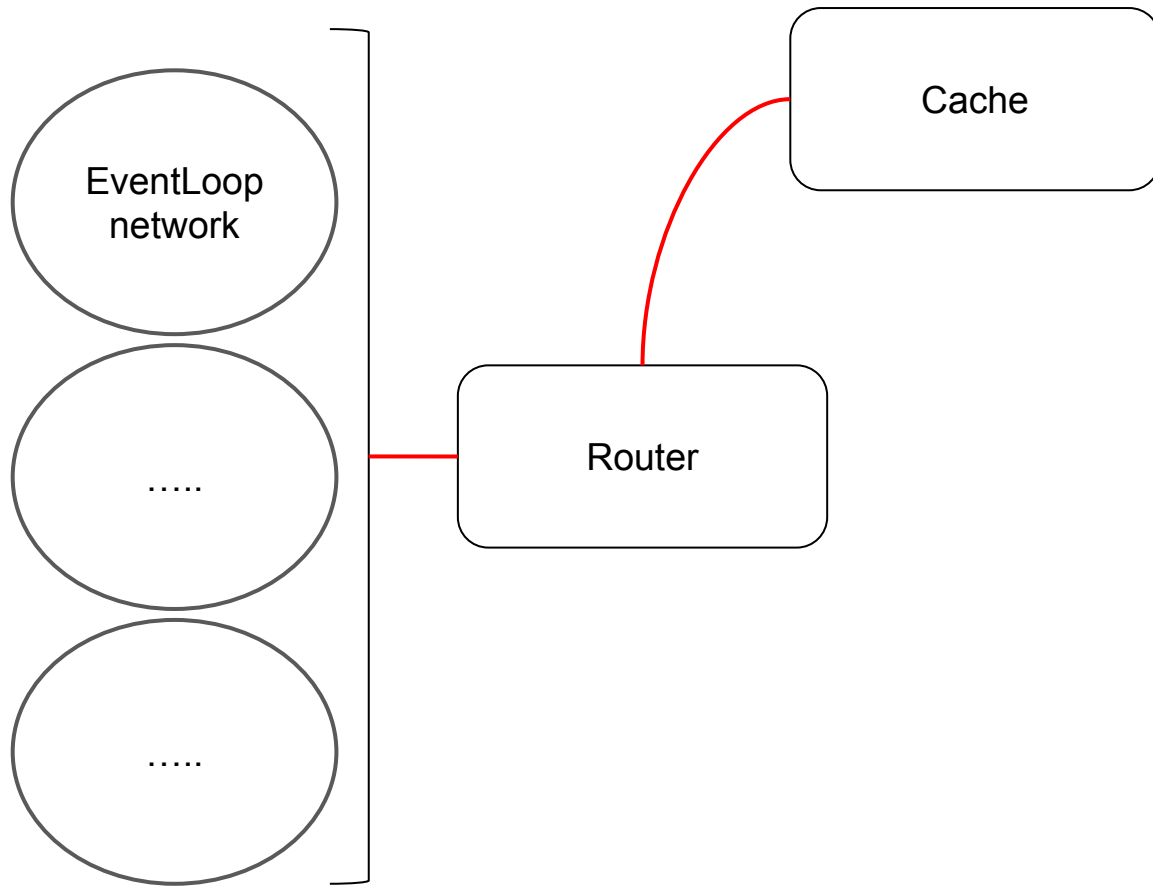
- threads + mutex +..... -> difficult path
- FSM -> What to take from the library ?
- Coroutines -> ? C++ 20 ?
- Actor-Model

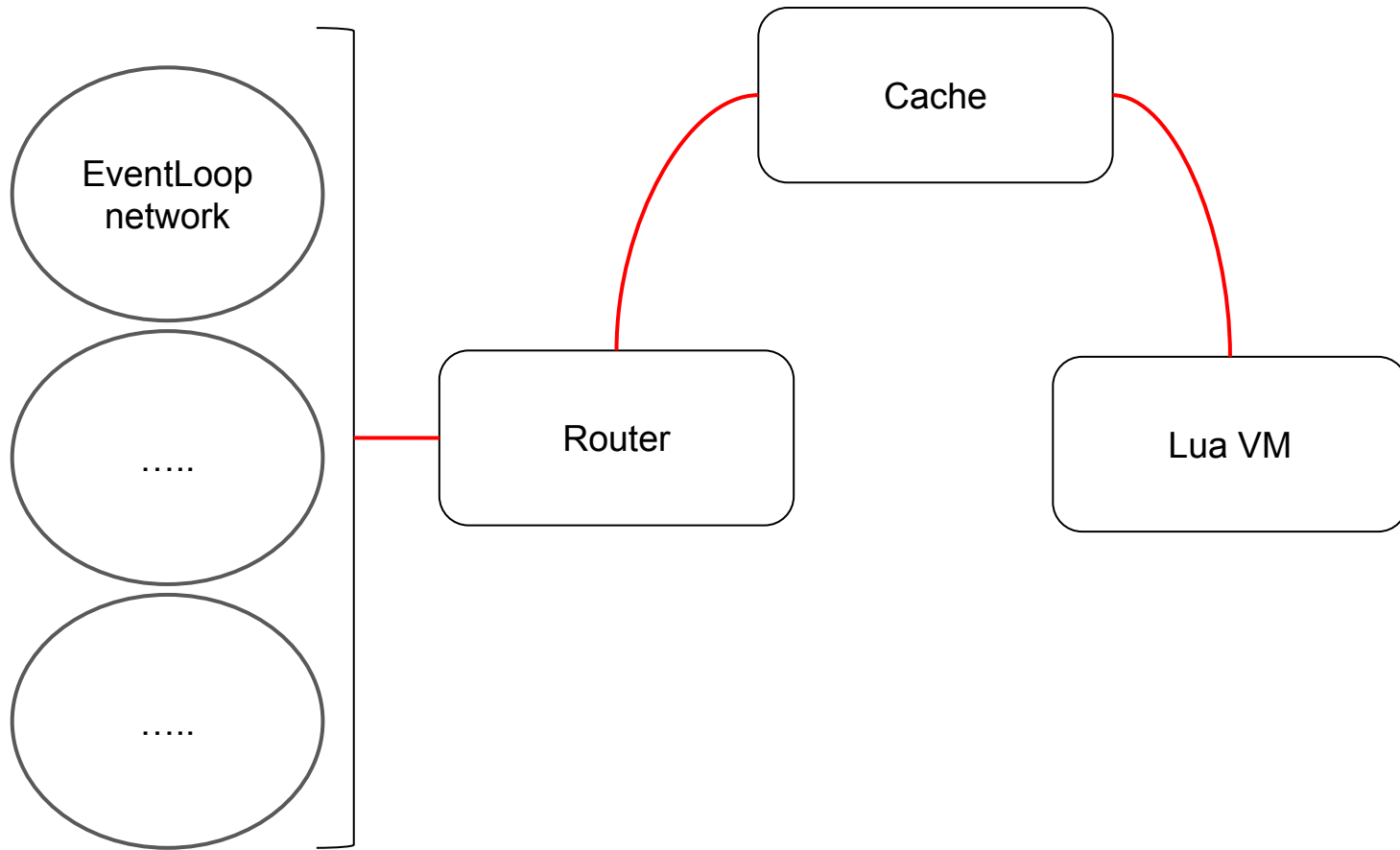
- threads + mutex +..... -> difficult path
- FSM -> What to take from the library ?
- Coroutines -> ? C++ 20 ?
- Actor-Model -> Favorite choice

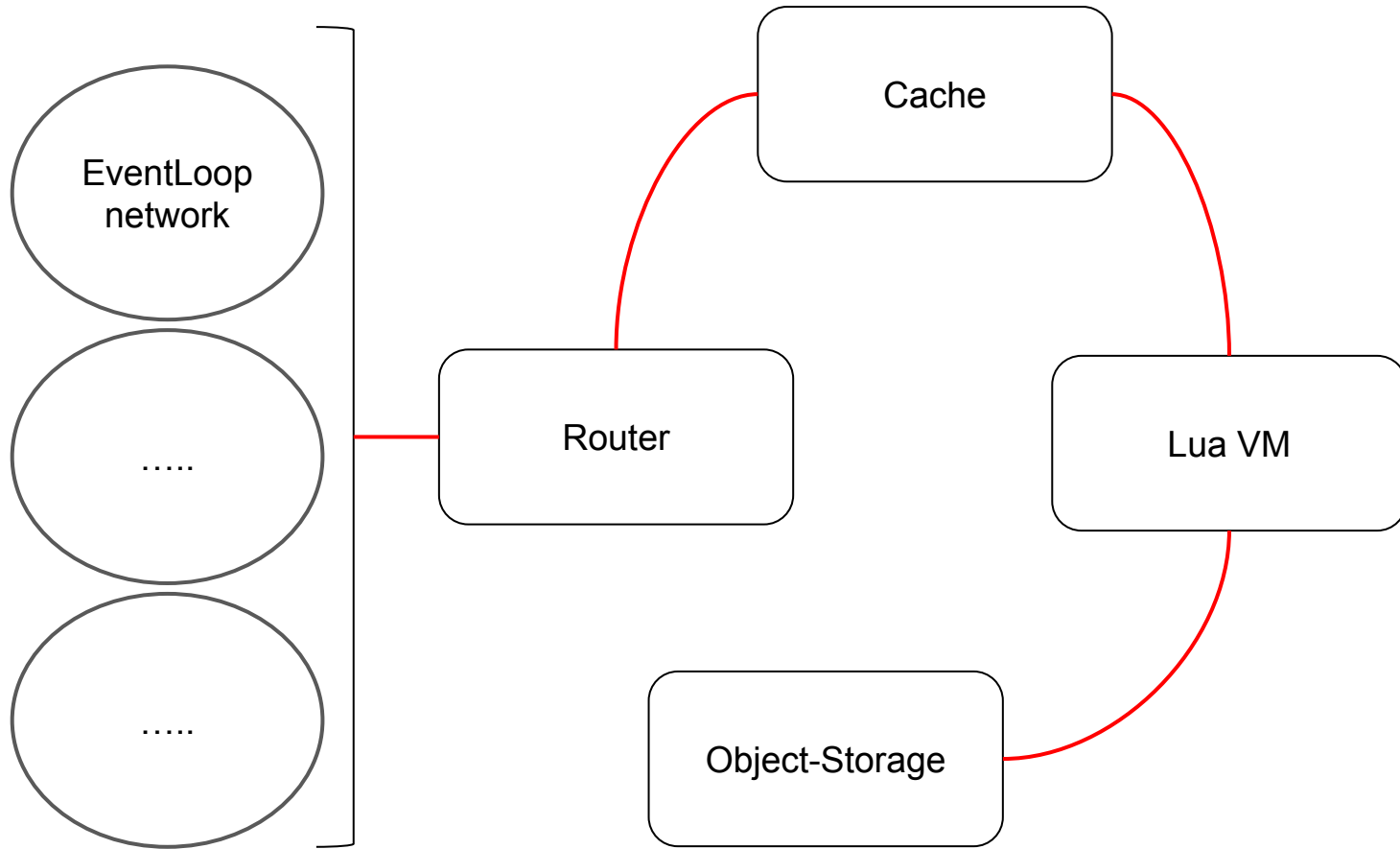
- threads + mutex +..... -> difficult path
- FSM -> What to take from the library ?
- Coroutines -> ? C++ 20 ?
- Actor-Model -> Favorite choice
 - message -> no implicit copying

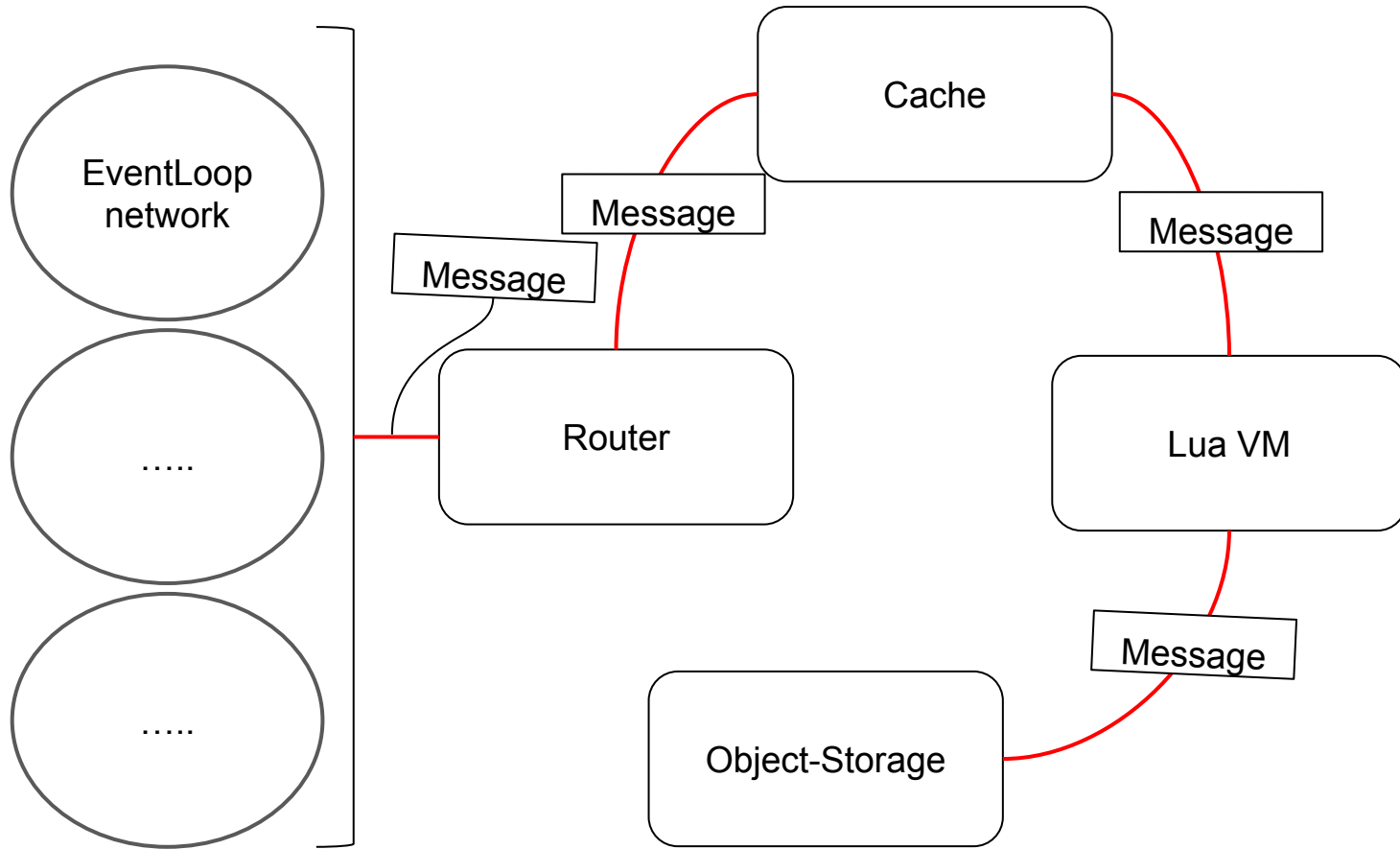


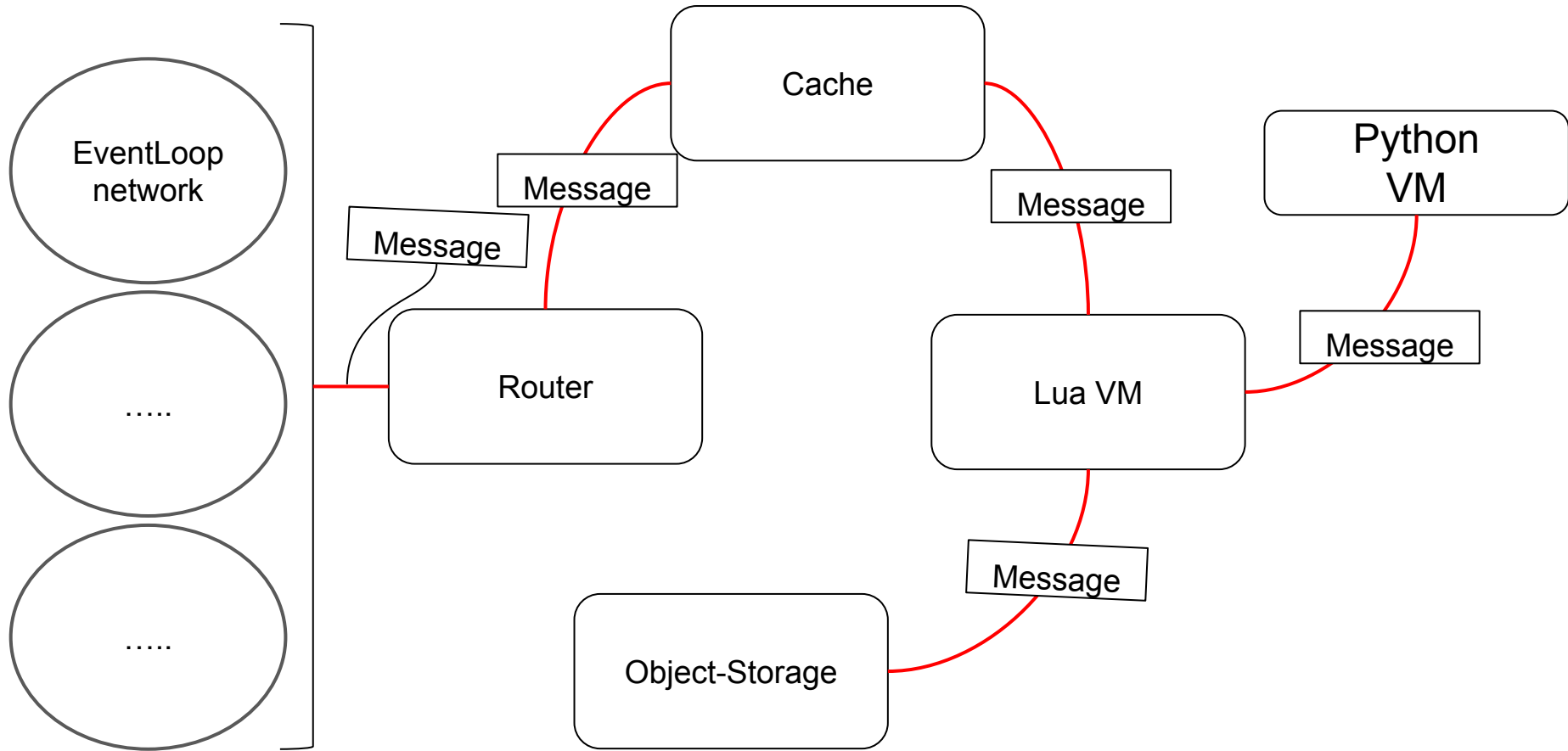












```
int main(int argc, char **argv) {  
    ///...  
    goblin_engineer::dynamic_environment env;  
    init_service(env);  
    env.initialize();  
    return env.startup();  
}
```

```
void init_service(goblin_engineer::dynamic_environment&env) {  
    ///...
```

```
    auto& lua = env.add_service<lua_engine::engine>();  
    auto& http = env.add_data_provider<http::server>(router);  
    http->add_shared(lua.address());  
    lua->add_shared(http.address());
```

```
    ///...
```

```
class async_actor {  
    ///...  
    virtual ~async_actor() {  
        ///...  
    }  
};
```

```
struct abstract_service: public actor_zeta::async_actor  
{  
  
    ~abstract_service() = default;  
  
    virtual void startup(goblin_engineer::context_t *) = 0;  
  
    virtual void shutdown() = 0;  
};
```



```
void init_service(goblin_engineer::dynamic_environment&env) {  
    ///...  
    auto& lua = env.add_service<lua_engine::engine>();  
    auto& http = env.add_data_provider<http::server>(router);  
    http->add_shared(lua.address());  
    lua->add_shared(http.address());  
  
    ///...  
}
```

```
void init_service(goblin_engineer::dynamic_environment&env) {  
    ///...  
    auto& lua = env.add_service<lua_engine::engine>();  
    auto& python = env.add_service<python_engine::engine>();  
    auto& http = env.add_data_provider<http::server>(router);  
    http->add_shared(lua.address()); ///... M:N  
    lua->add_shared(http.address());  ///... M:N  
  
    ///...  
}
```

```
void init_service(goblin_engineer::dynamic_environment&env) {  
    ///...  
    auto& lua = env.add_service<lua_engine::engine>();  
    auto& python = env.add_service<python_engine::engine>();  
    auto& http = env.add_data_provider<http::server>(router);  
    http->add_shared(lua.address()); ///... M:N  
    lua->add_shared(http.address()); ///... M:N  
    lua->join(python.address()); ///... M:1  
    ///...  
}
```

```
namespace py = pybind11;
class python_engine final : public
goblin_engineer::abstract_service {
    ///...
    ~lua_engine() override;

    void startup(goblin_engineer::context_t *) override;

    void shutdown() override;
    ///...
private:
    py::module pyrocketjoe
    wrapper_ptr<std::thread> executor;
};
```

```
namespace py = pybind11;
class python_engine final : public
goblin_engineer::abstract_service {
    ///...
    ~lua_engine() override;

    void startup(goblin_engineer::context_t *) override;

    void shutdown() override;
    ///...
private:
    [py::module pyrocketjoe]
    wrapper_ptr<std::thread> executor;
};
```

```
pyrocketjoe.def(  
    "smart_open",  
    [this](std::string &filename, std::string &mode) {  
        ///...  
    }  
    ///...  
);
```

```
executor = std::make_unique<std::thread>(
    [this]() {
        ///...
    auto locals = py::dict("path"_a=path_script, "pyrocketjoe"_a=pyrocketjoe);
        py::exec(R"(
            import sys, os
            from importlib import import_module
            sys.modules['pyrocketjoe'] = pyrocketjoe
            sys.path.insert(0, os.path.dirname(path))
            module_name, _ = os.path.splitext(path)
            import_module(os.path.basename(module_name))
        )", py::globals(), locals);
        ///...
    }
);
```

Examples of relationships


```
#!/usr/bin/env python3
#...
from aiohttp.web import run_app
#...

def main() -> None:
    #...
    run_app(init(config), host=config.app_host, port=config.app_port)

if __name__ == '__main__':
    main()
```

```
async def init(config: Config) -> Application:  
    #...  
    with open("big_file.txt", "r") as file:  
        data = file.readline()  
        data_storage.update(transform(data))  
    #...  
    return app
```

```
#...1m - 15m
async def init(config: Config) -> Application:
    #...
    with open("big_file.txt", "r") as file:
        data = file.readline()
        data_storage.update(transform(data))
    #...
    return app
```

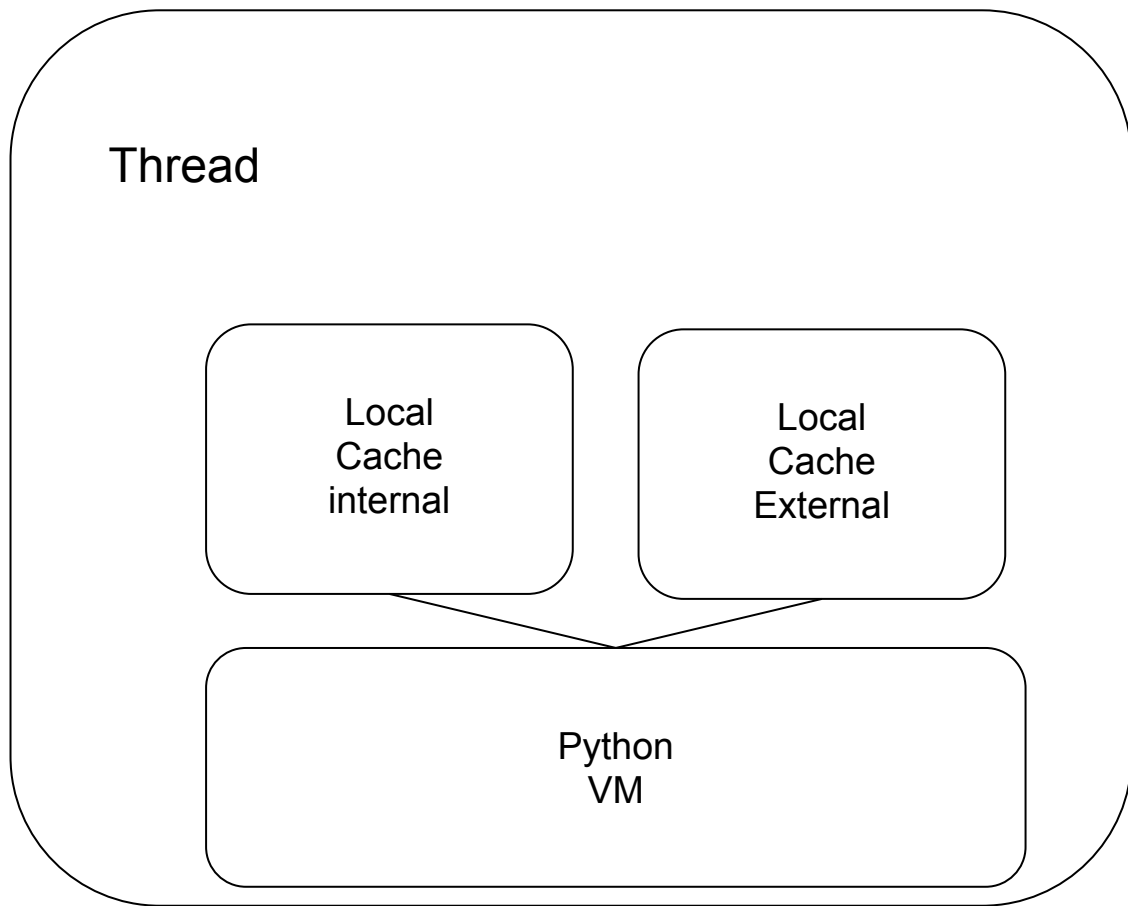
```
from pyrocketjoe import smart_open

async def init(config: Config) -> Application:
    #...
    with smart_open("big_file.txt", "r") as file:
        data = file.readline()
        data_storage.update(transform(data))
    #...
    return app
```

```
from pyrocketjoe import smart_open

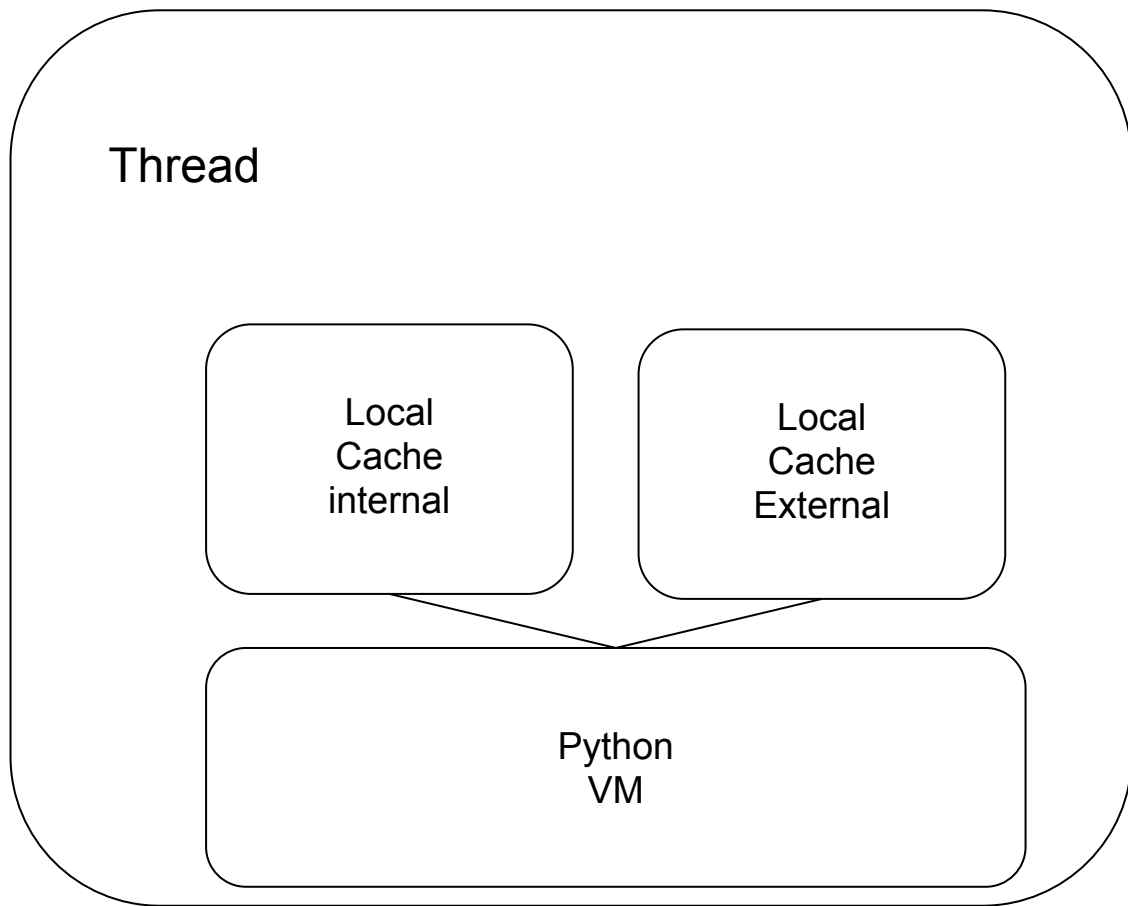
async def init(config: Config) -> Application:
    #...
    with smart_open("big_file.txt", "r") as file:
        data = file.readline()
        data_storage.update(transform(data))
    #...
    return app
```

Standalone Thread Python VM

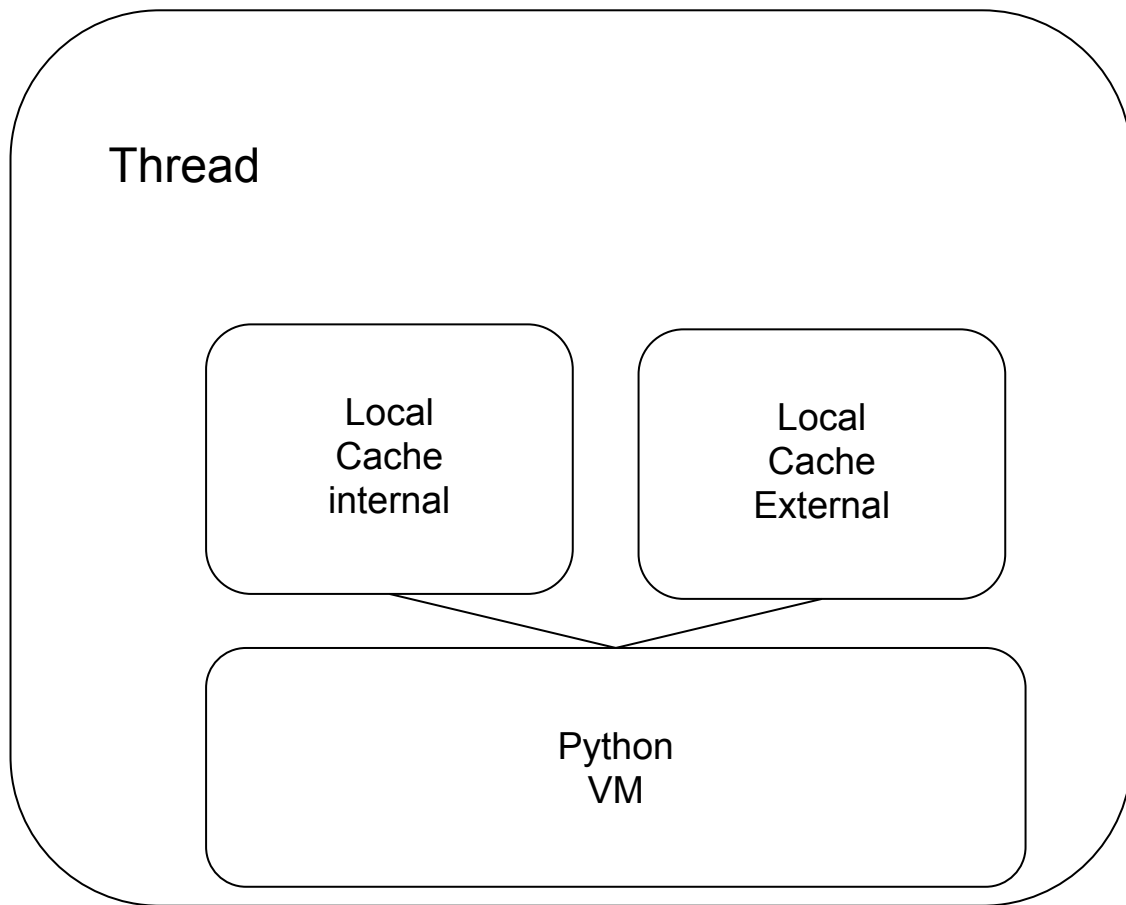
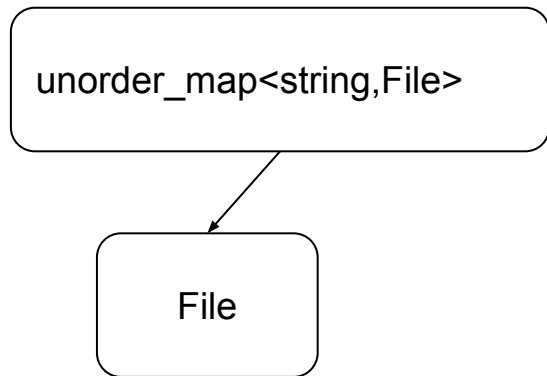


Standalone Thread Python VM

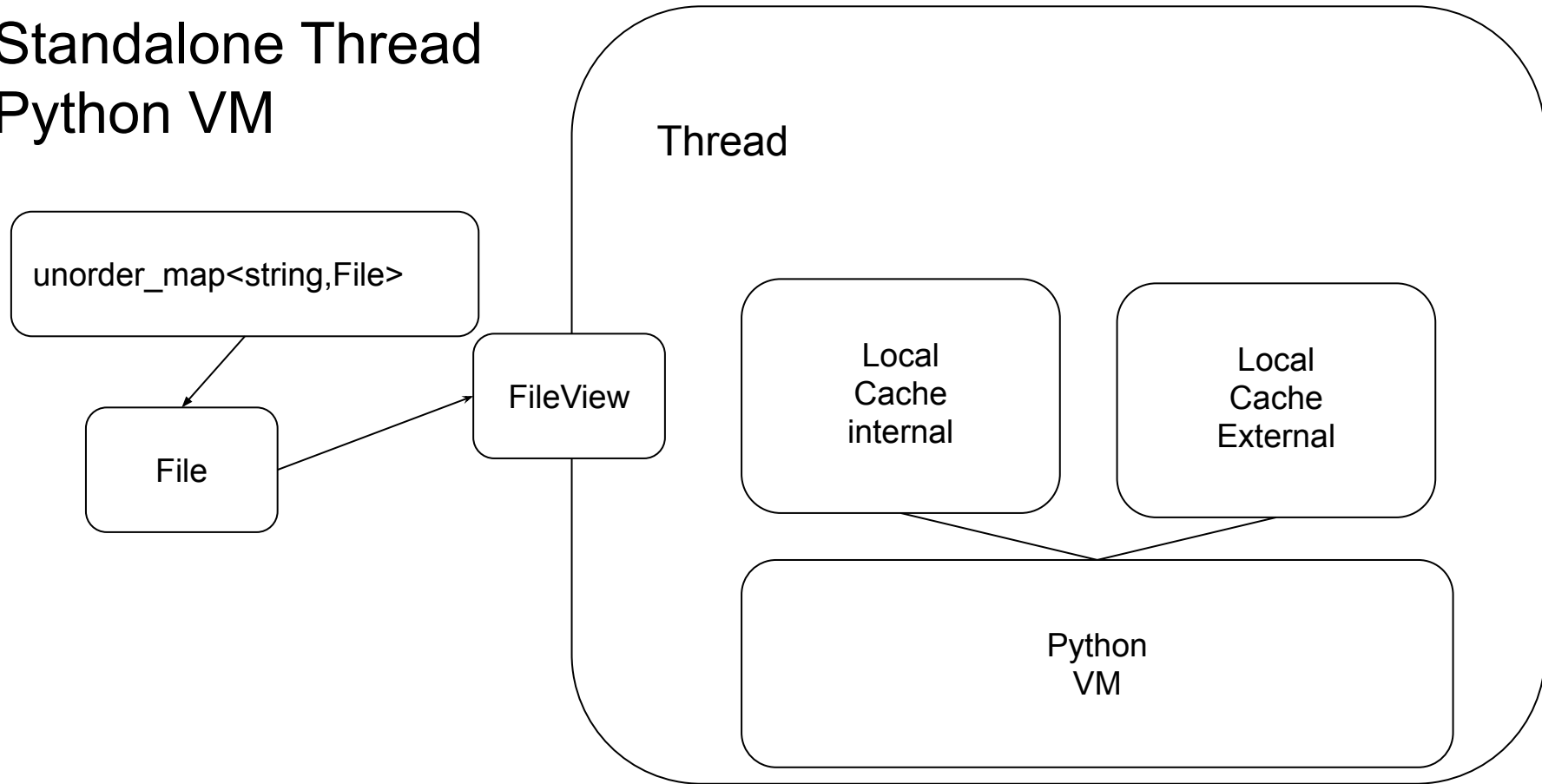
`unordered_map<string,File>`



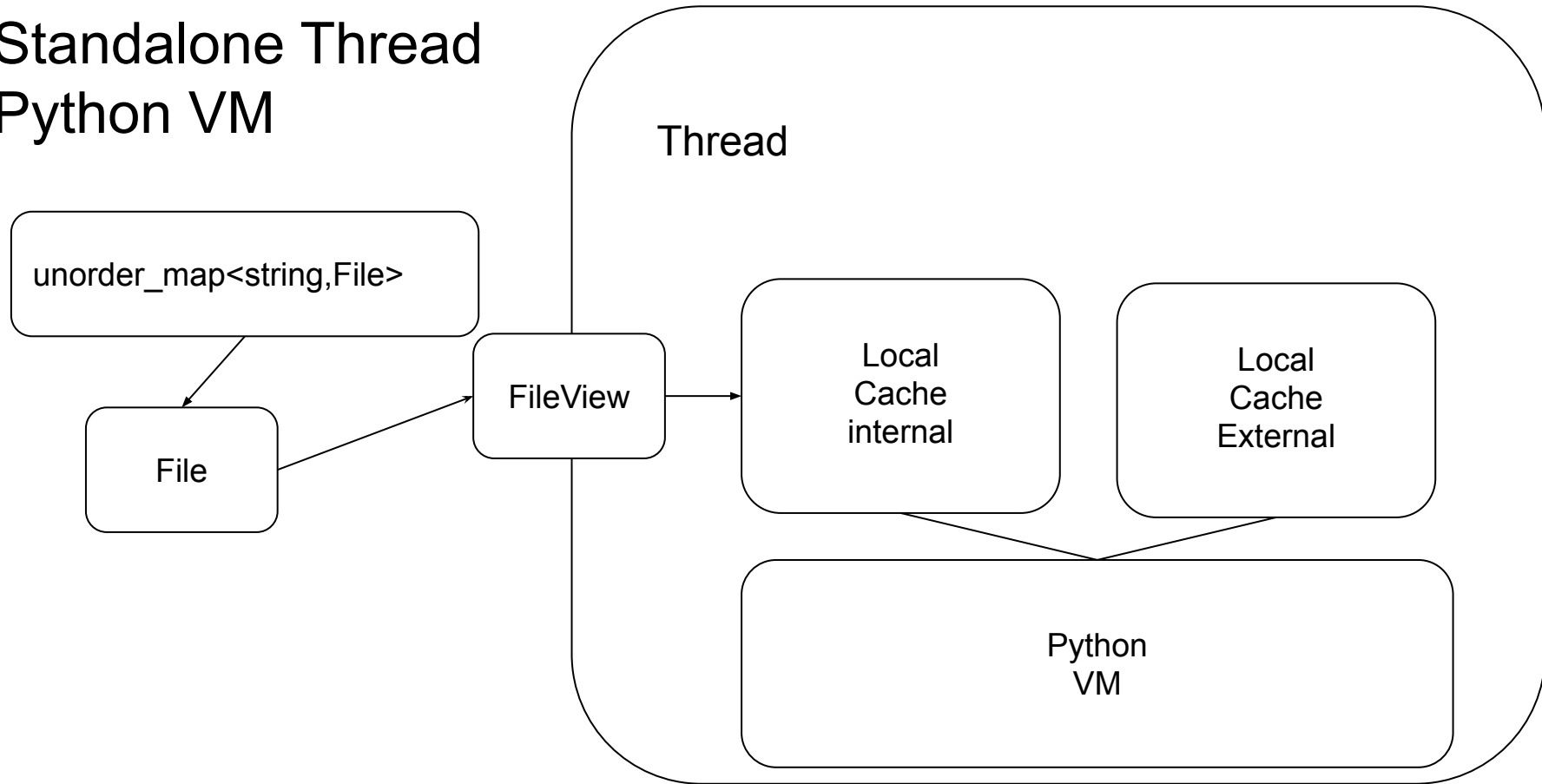
Standalone Thread Python VM



Standalone Thread Python VM



Standalone Thread Python VM



Выводы:

- StartUp : 1m - 15m
- StartUp : 1m - 5m

Выводы:

- StartUp : 1m - 15m
- Python

- StartUp : 1m - 5m
- Python + C++ extension

Выводы:

- StartUp : 1m - 15m
- Python
- Number of machines in Analytics: 100

- StartUp : 1m - 5m
- Python + C++ extension
- Number of machines in Analytics: 20
- Number of machines in Analytics + Product: 50

Basic library

spdlog

Rapidjson

Boost.asio

Boost.beast

pybind11

yaml-cpp

sol2 -> 3

cxxopts

conan

Basic library

spdlog

Rapidjson

Boost.asio

Boost.beast

pybind11

yaml-cpp

sol2 -> 3

cxxopts

conan

Basic components

Actor-Zeta

goblin-engineer

Basic library

spdlog

Rapidjson

Boost.asio

Boost.beast

pybind11

yaml-cpp

sol2 -> 3

cxxopts

conan

Basic components

Actor-Zeta

goblin-engineer

Open Source Prototype

RocketJoe

Borgardt Alexander

aa.borgardt@yandex.ru



kotbegemot

<https://github.com/jinntechio>

<https://github.com/blizgard>

<https://github.com/smart-cloud/RocketJoe>