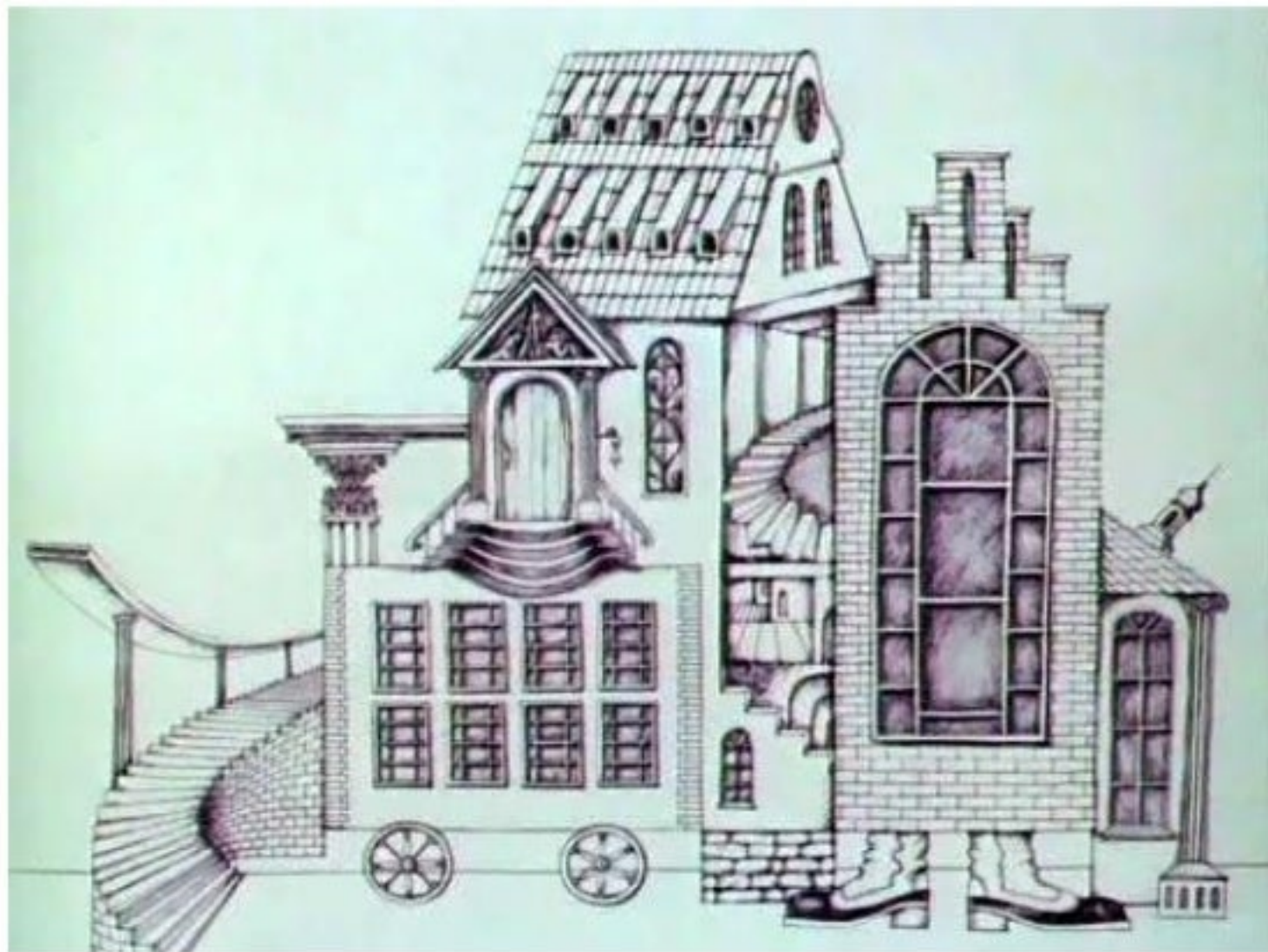




Integrating python to enable Analytics for C++

Borgardt Alexander

199x



20xx-2005



2006-2013



online
Shop
0

online
Shop
...

online
Shop
N

online
Shop
0

online
Shop
...

online
Shop
N

Platform

online
Shop
0

online
Shop
...

online
Shop
N

Platform

online Shop 0 / Front-end

online Shop 0 / Front-end

```
graph TD; A[online Shop 0 / Front-end] --- B(Nginx);
```

The diagram consists of a large rectangular box at the top containing the text "online Shop 0 / Front-end". A vertical line extends from the bottom center of this box to a smaller, rounded rectangular box below it, which contains the text "Nginx".

Nginx

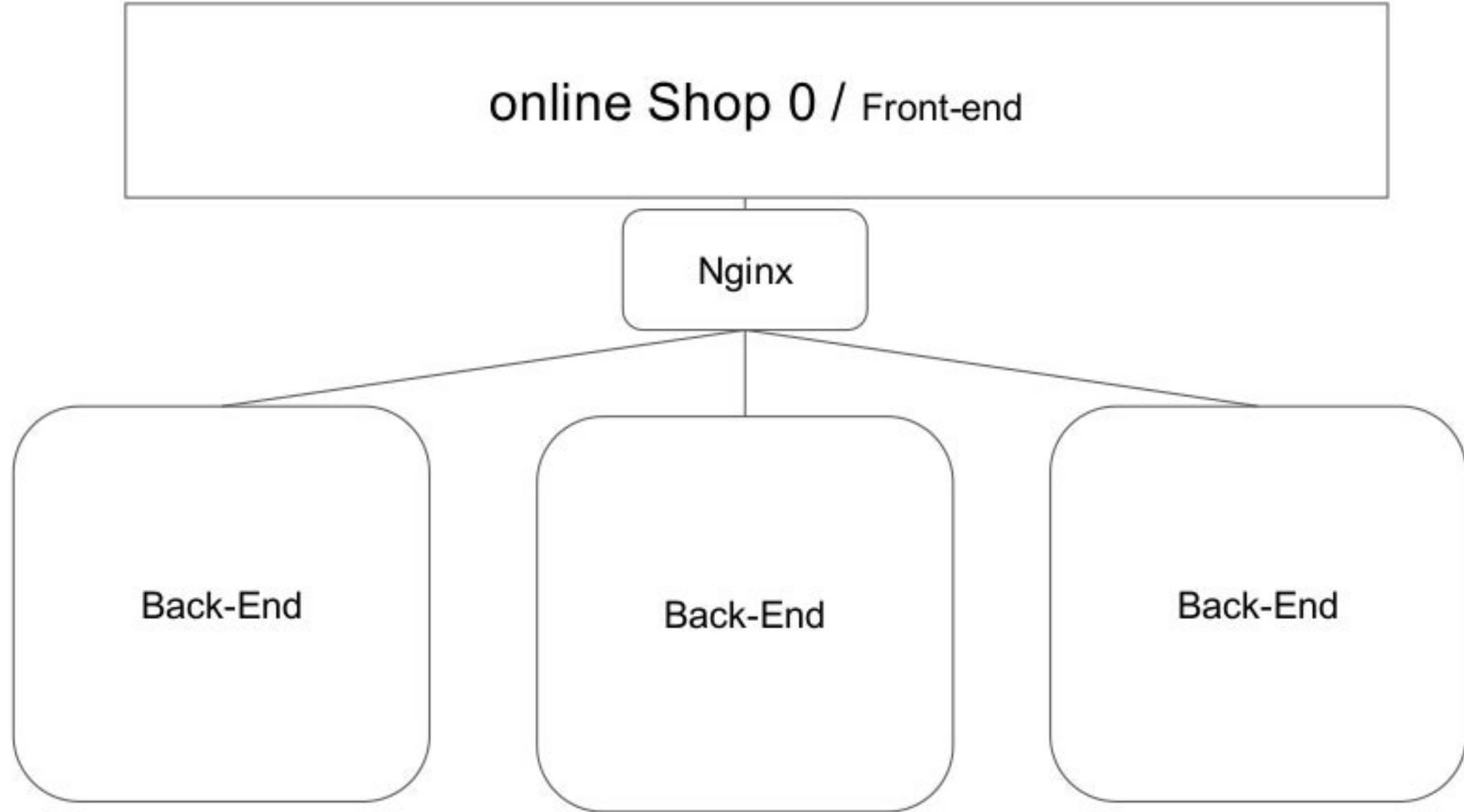
online Shop 0 / Front-end

Nginx

Back-End

Back-End

Back-End



online Shop 0 / Front-end

Nginx

Back-End

Back-End

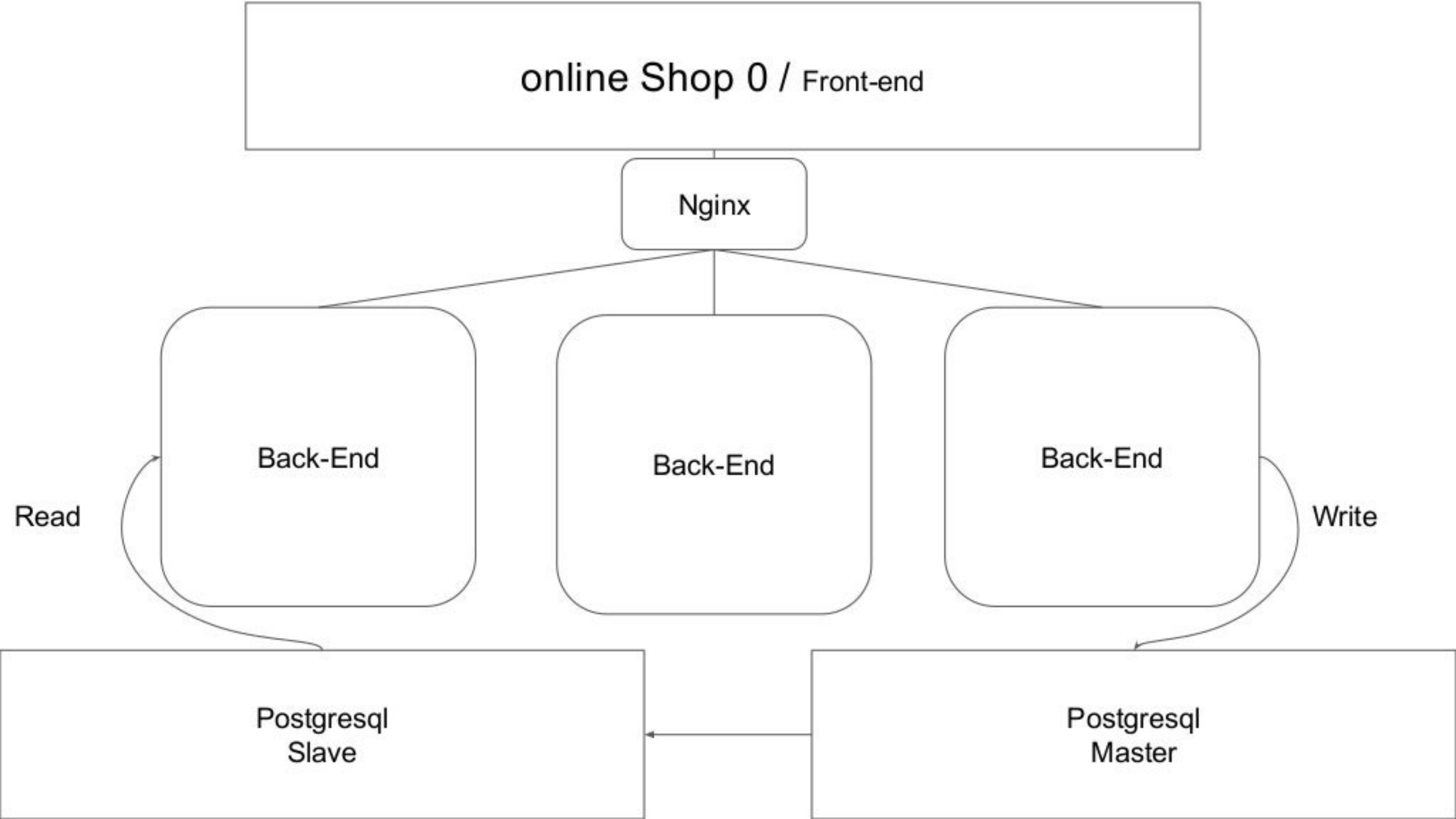
Back-End

Read

Write

Postgresql
Slave

Postgresql
Master



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 ?

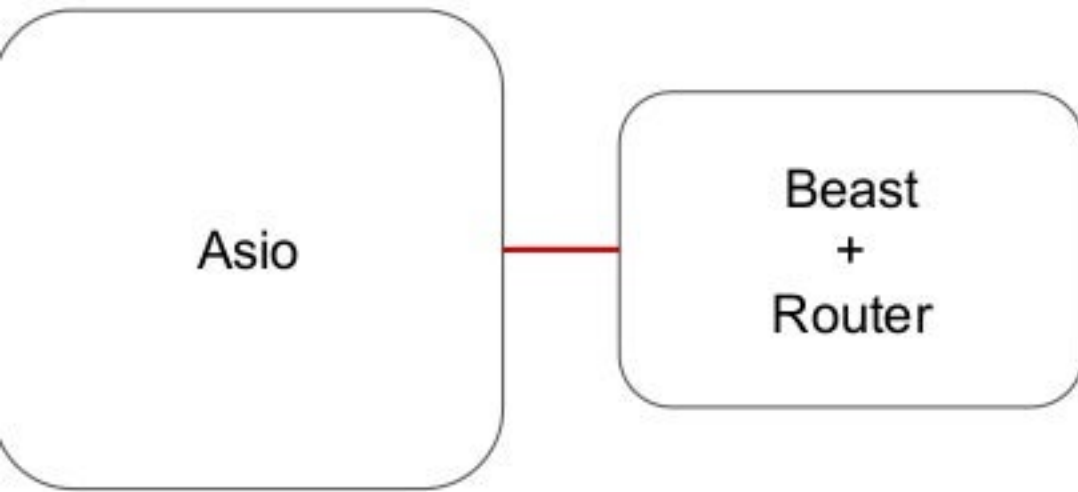
Actor-Model

Back-End

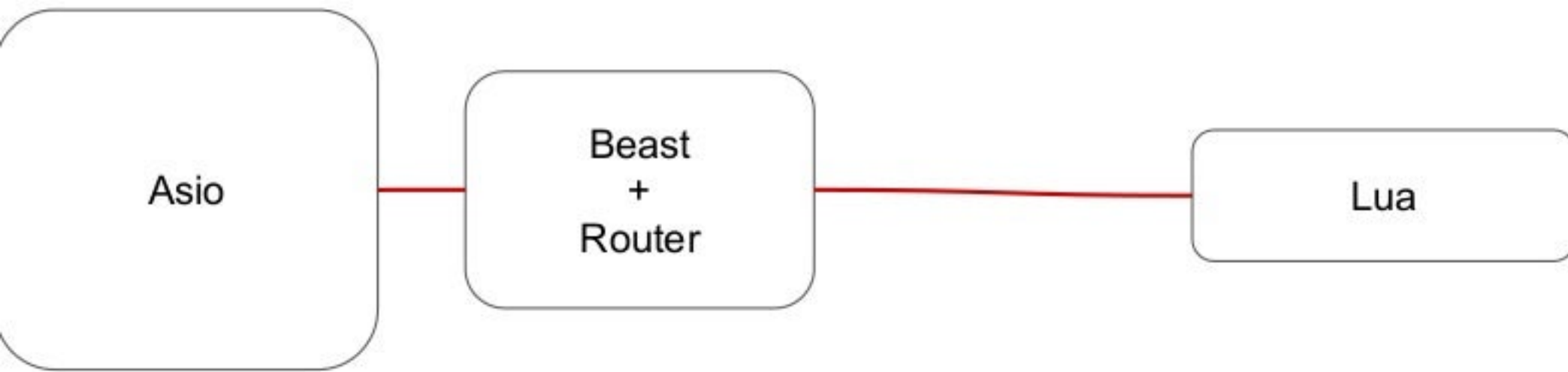


Asio

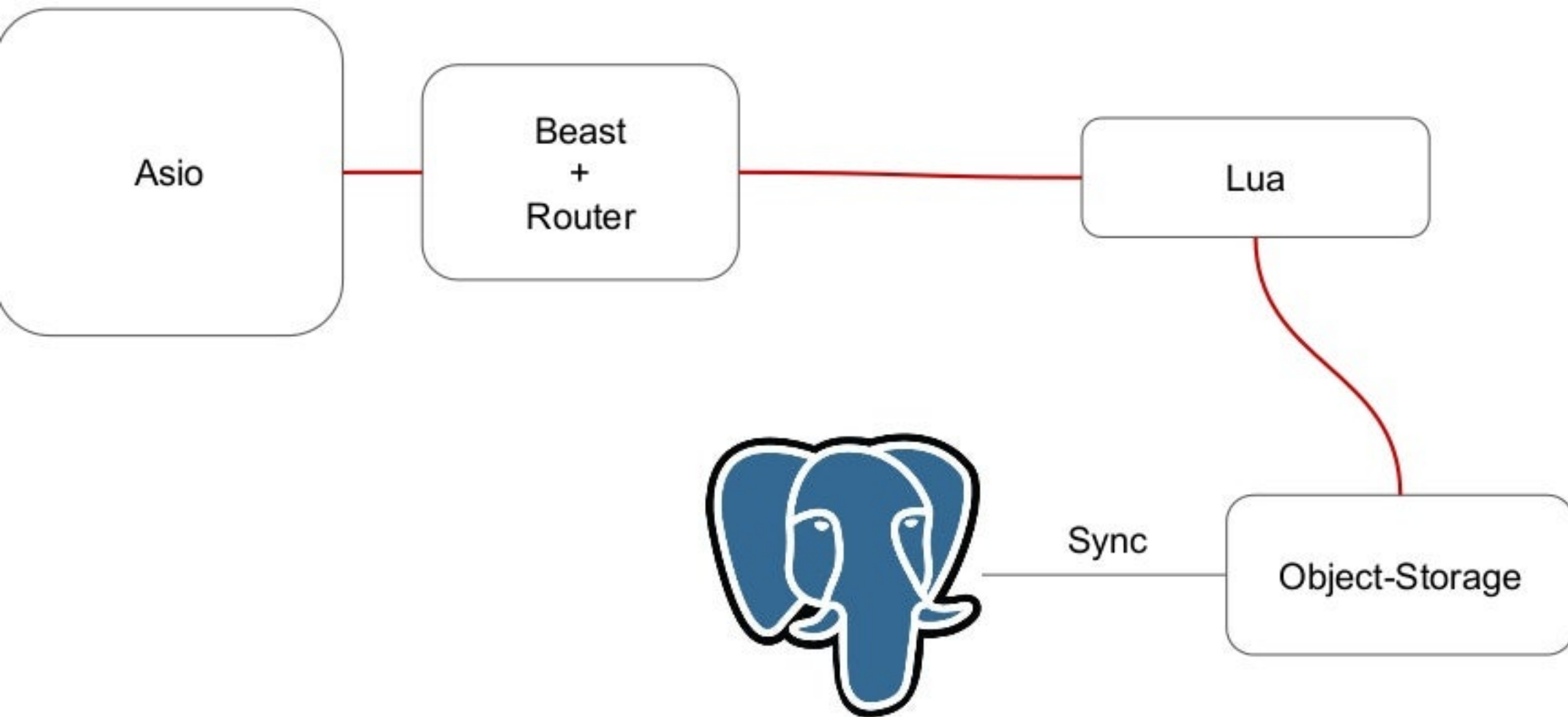
Back-End



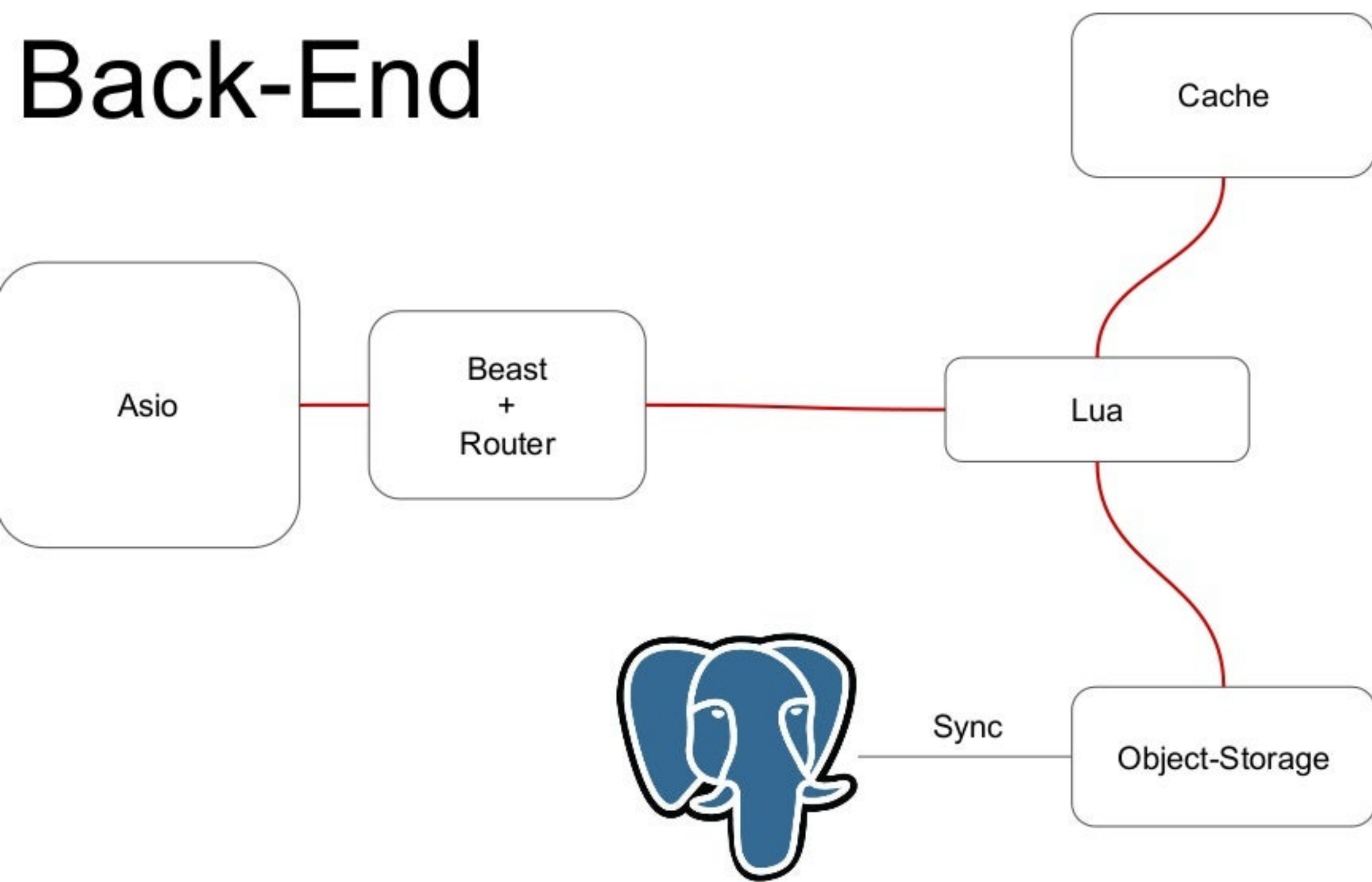
Back-End

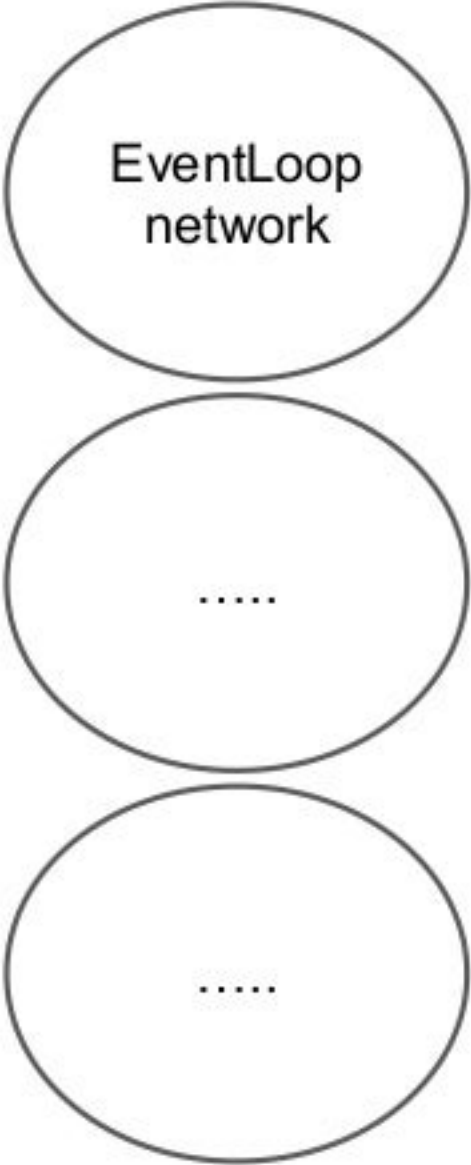


Back-End



Back-End



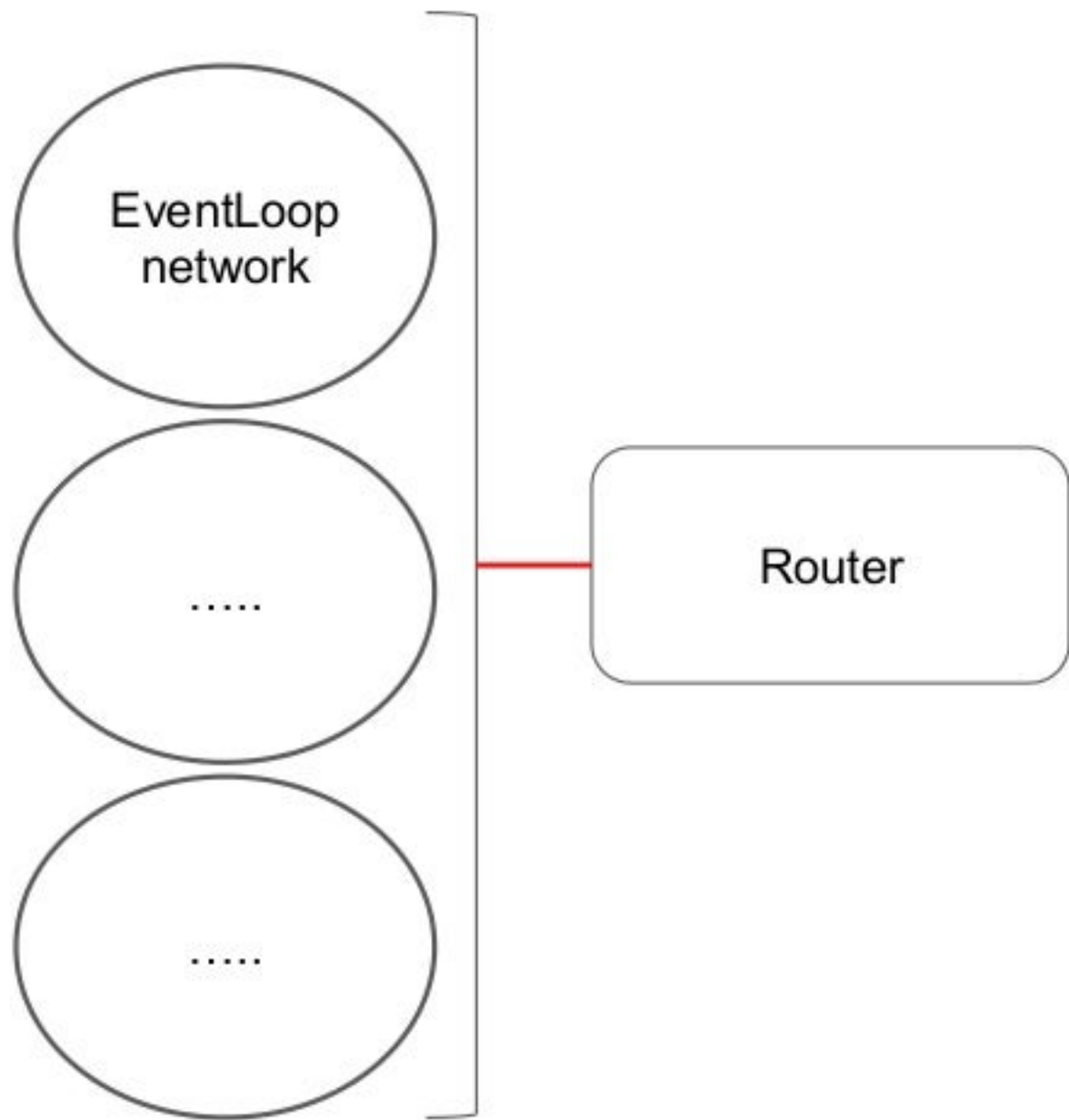


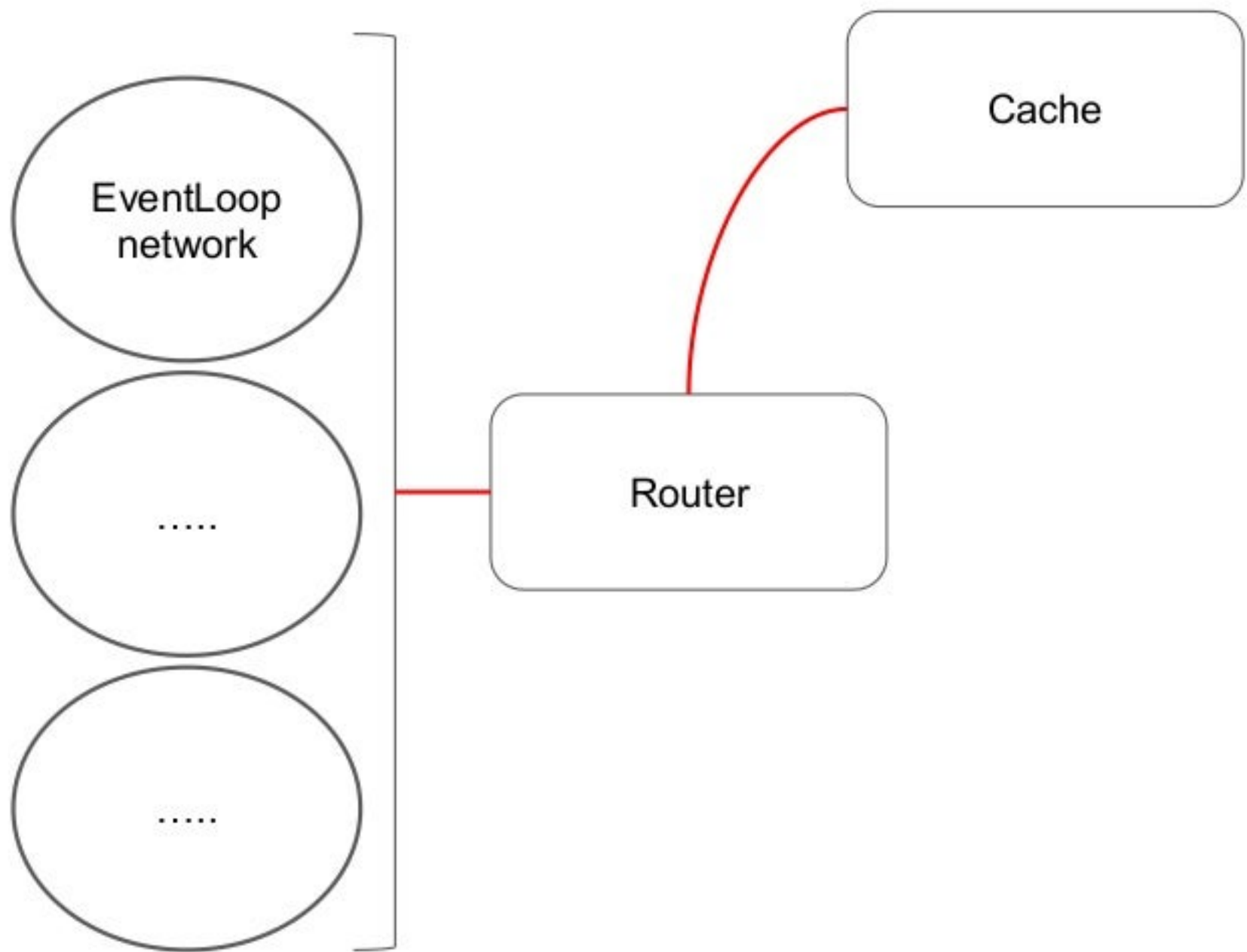
The diagram consists of three identical circles stacked vertically on the left side of the image. The top circle contains the text 'EventLoop' and 'network' on two lines. The middle and bottom circles each contain a horizontal ellipsis '.....' in the center.

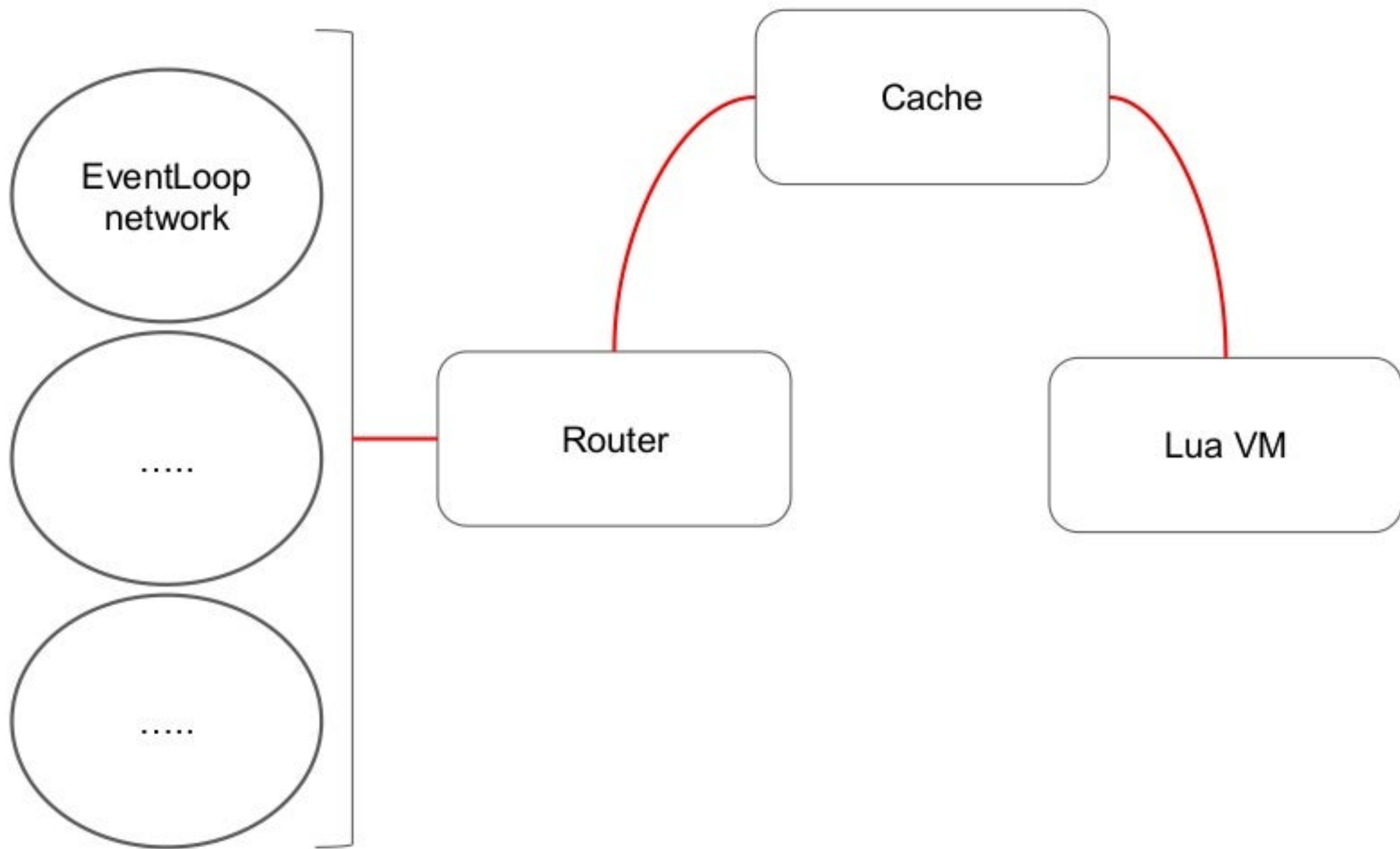
EventLoop
network

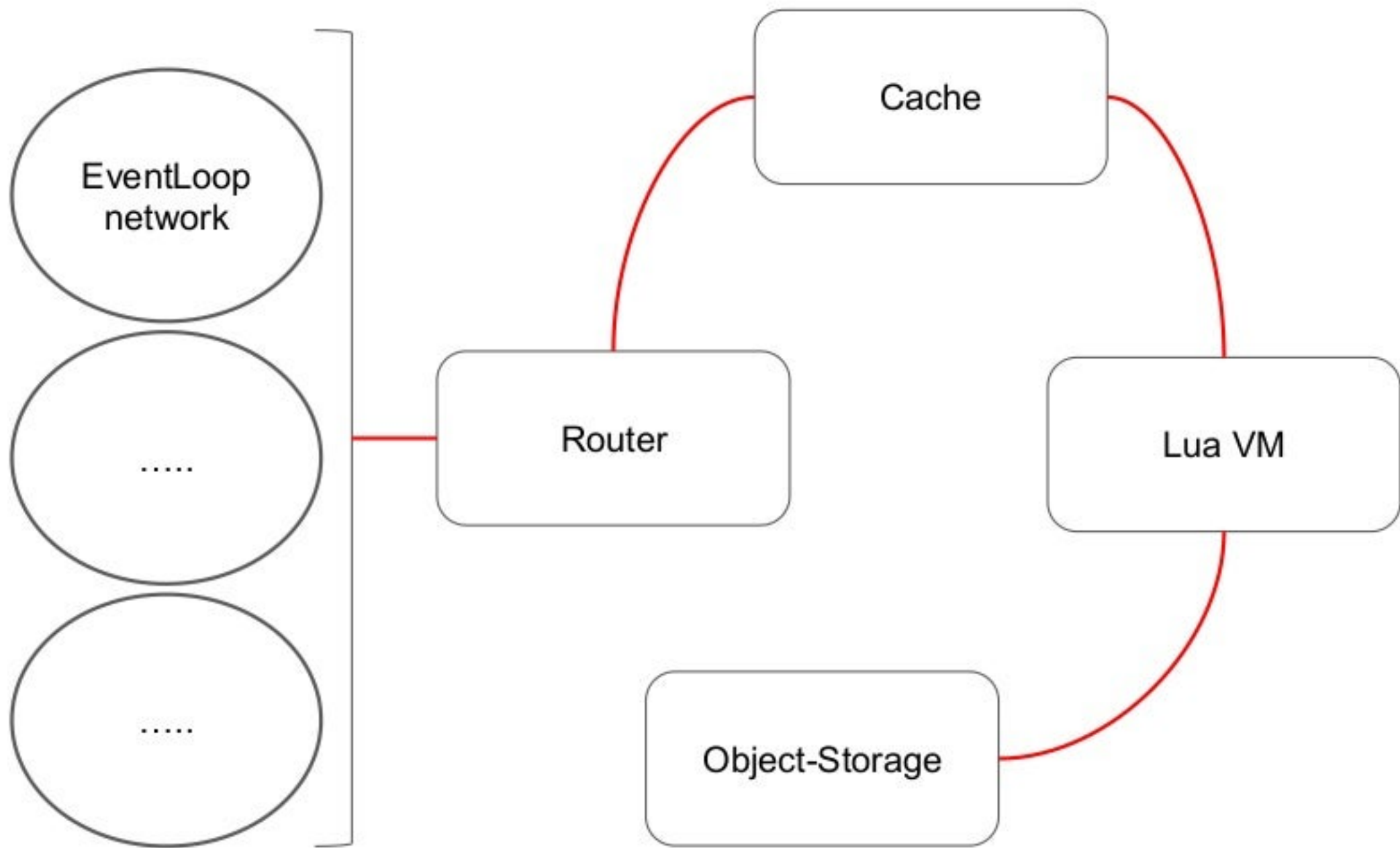
.....

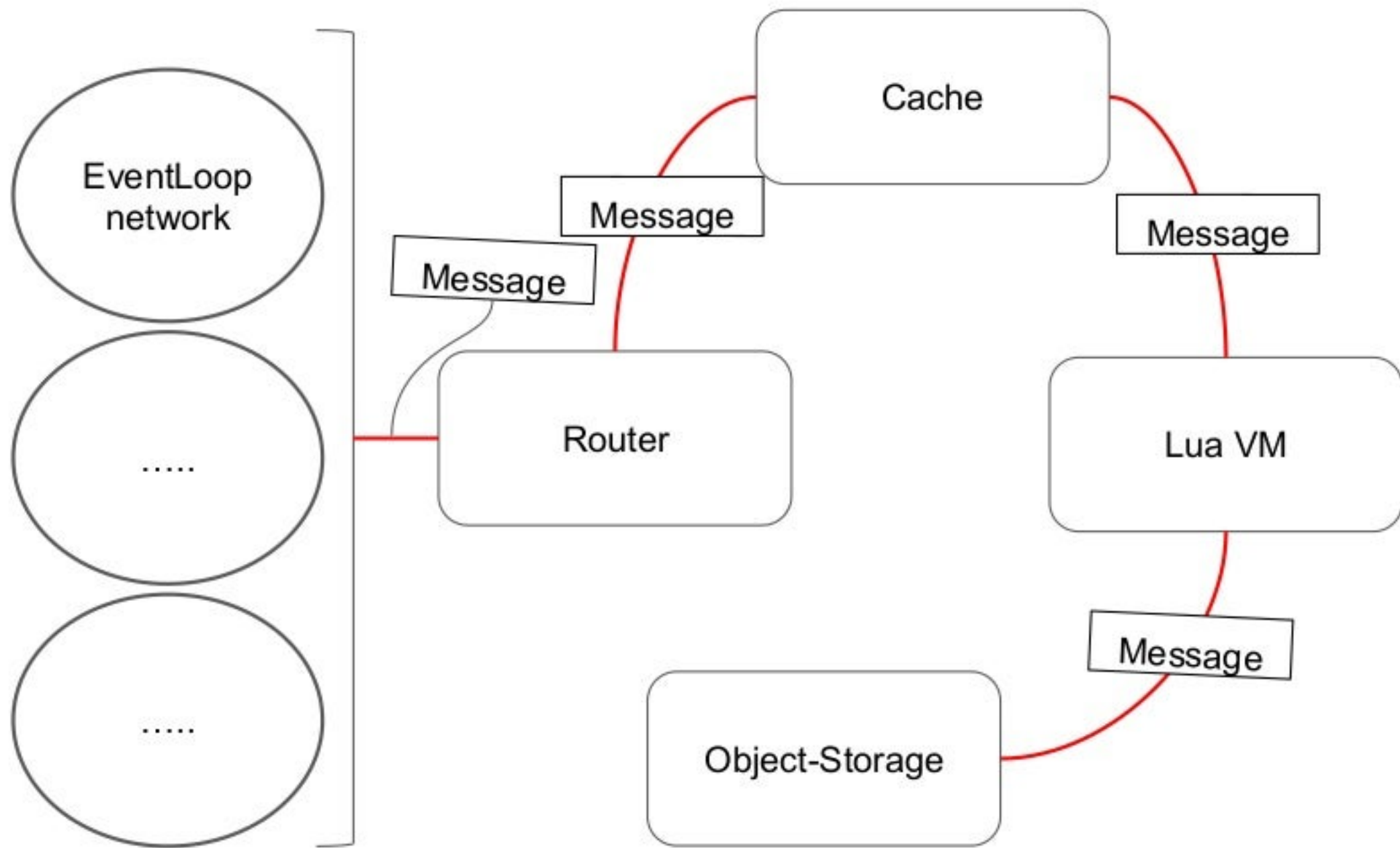
.....











Lua
VM

Lua VM

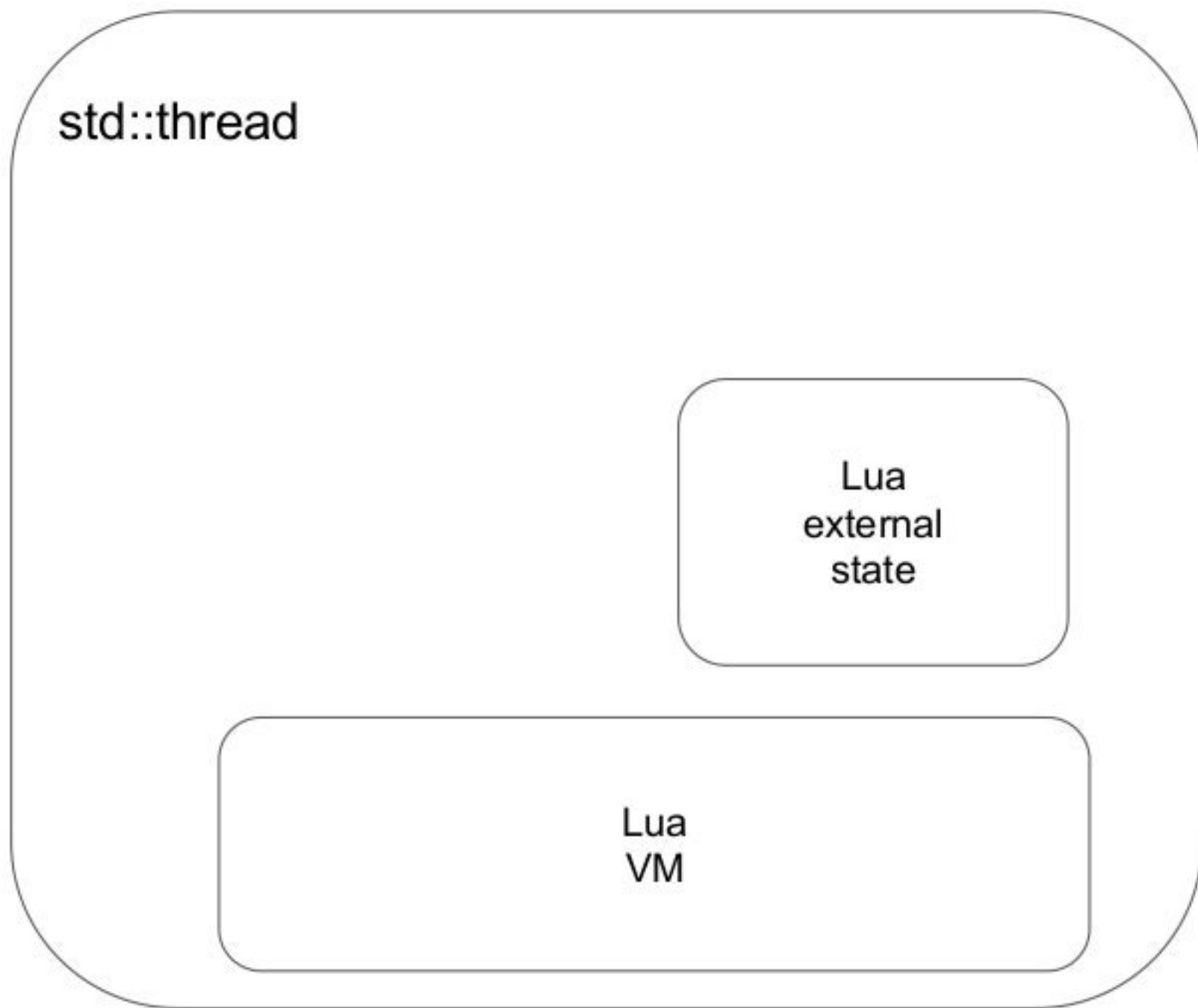
std::thread

Lua VM

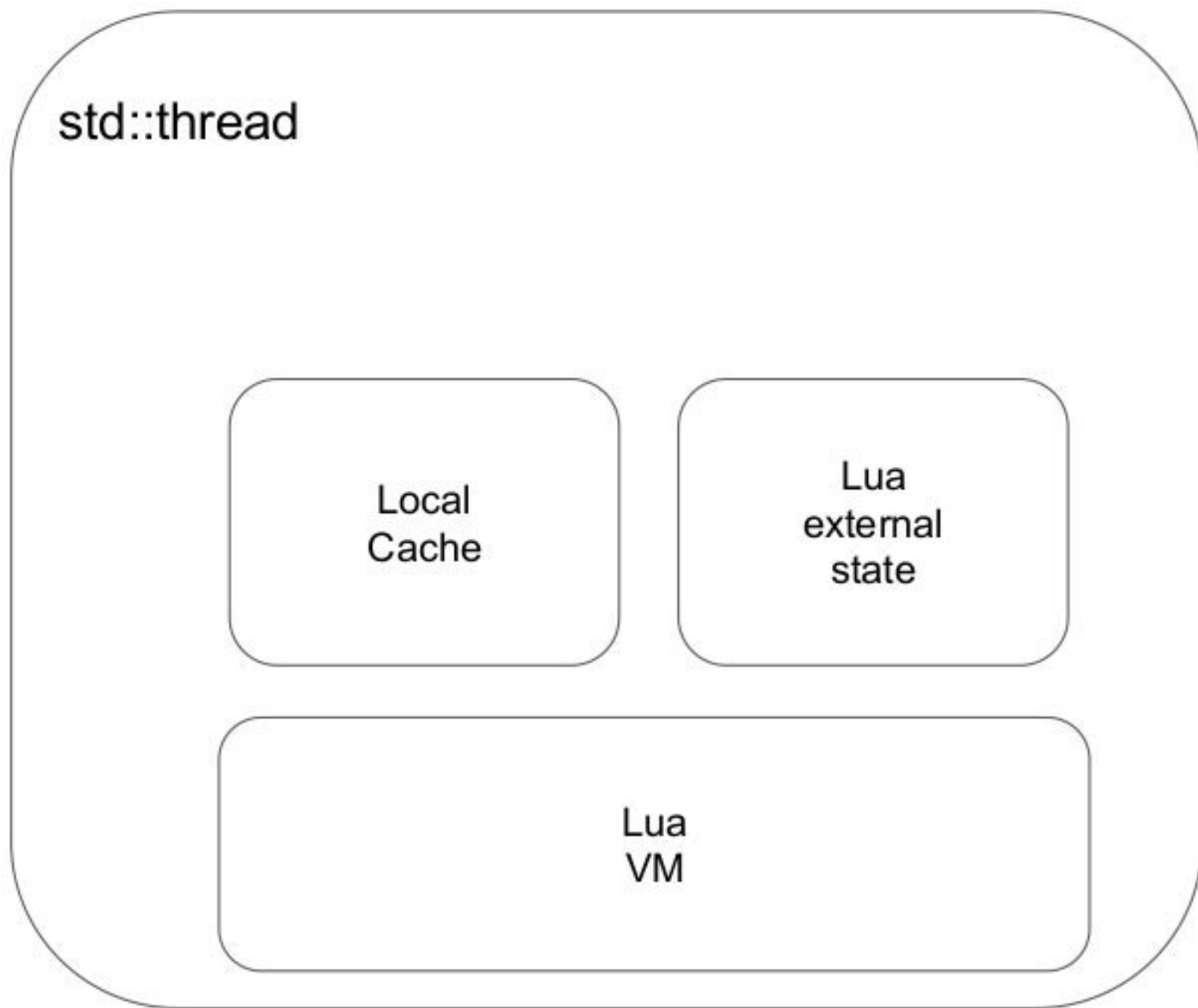
std::thread

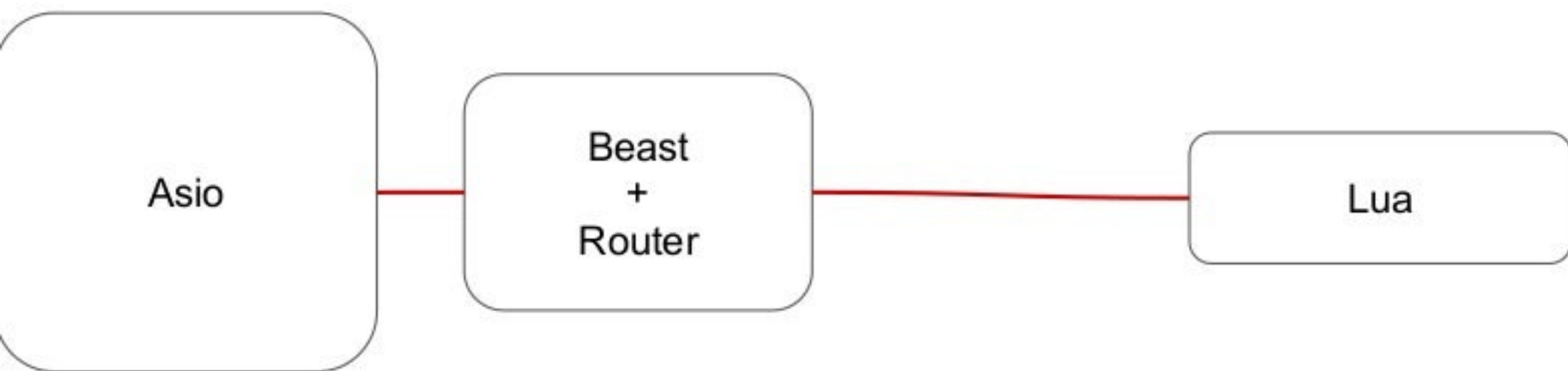
Lua
VM

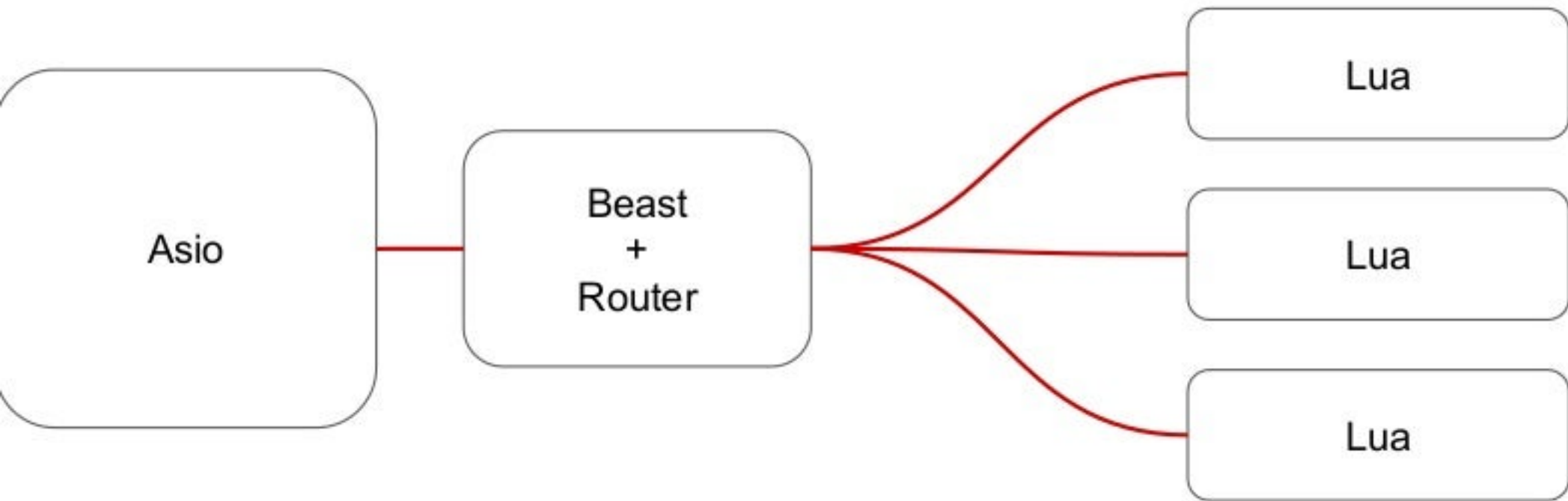
Lua VM

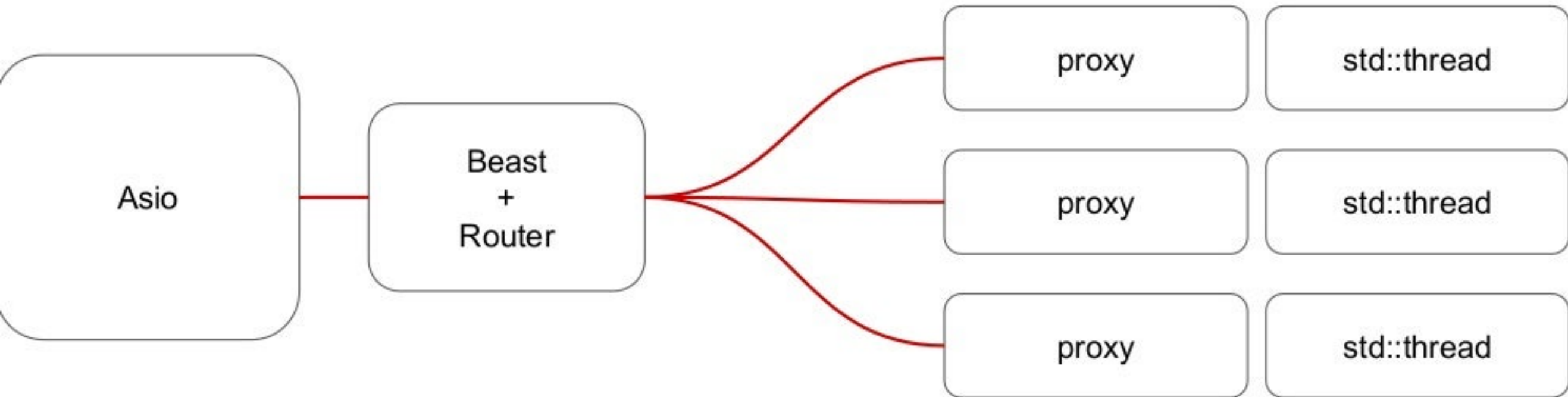


Lua VM









```
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;

};
```

```
class lua_engine final : public
goblin_engineer::abstract_service {
    ///...
    ~lua_engine() override;

    void startup(goblin_engineer::context_t *) override;

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

```
exuctor = std::make_unique<std::thread>(
    [this]() {
        auto script = lua.load_file(this->path_script);
        ///...
        script();
        ///...
    }
);
```

online
Shop
0

online
Shop
...

online
Shop
N

Platform

analytics

online
Shop
0

online
Shop
...

online
Shop
N

Platform

Примеры механик аналитиков.

Примеры механик аналитиков.

- Оценка качества закупленного трафика

Примеры механик аналитиков.

- Оценка качества закупленного трафика
- Рекомендации пользователя для пользователей

online Shop 0 / Front-end

```
graph TD; A[online Shop 0 / Front-end] --- B(Nginx)
```

A diagram showing a connection between a front-end application and a reverse proxy. A rectangular box at the top contains the text 'online Shop 0 / Front-end'. A vertical line descends from the center of this box to a smaller, rounded rectangular box below it, which contains the text 'Nginx'.

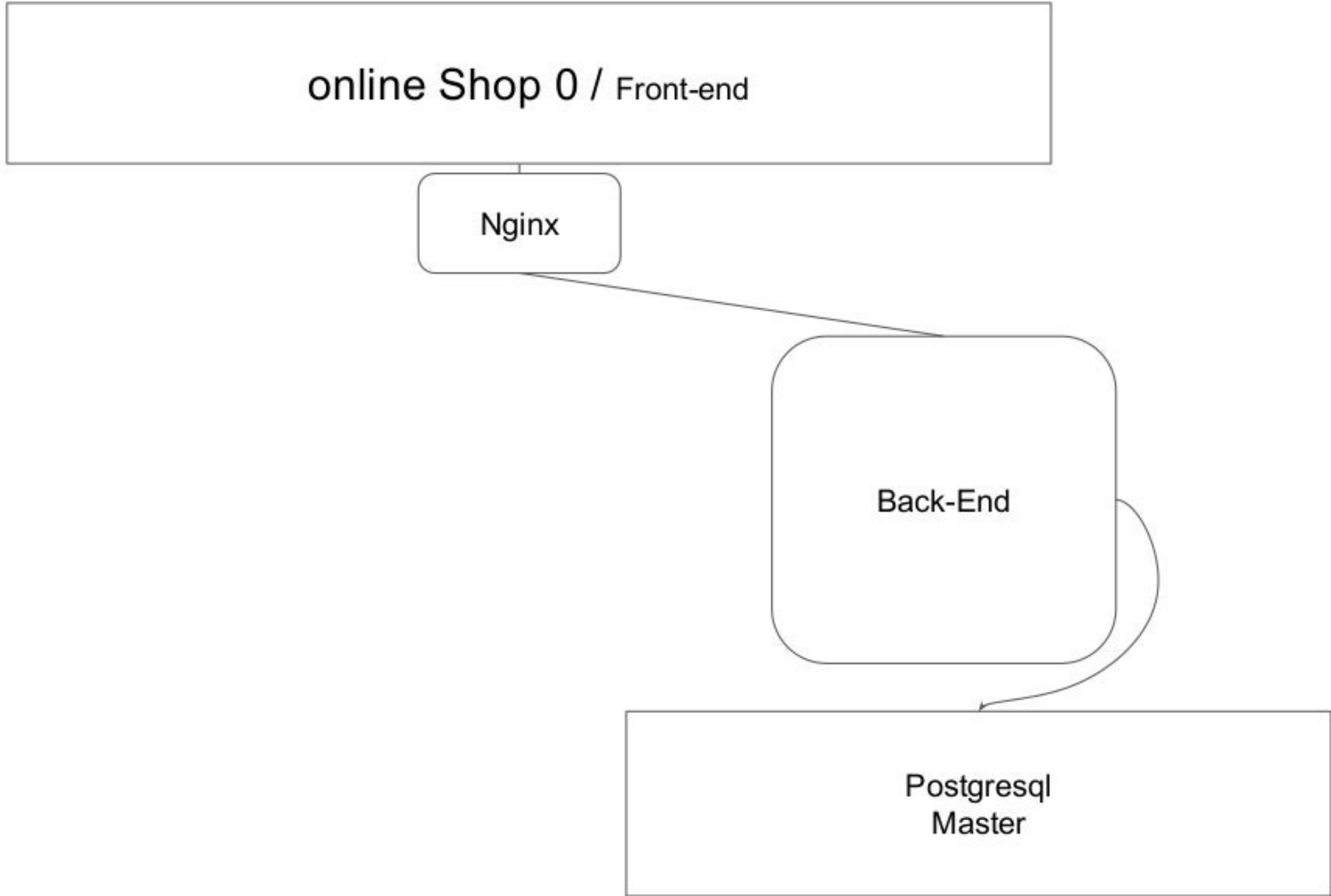
Nginx

online Shop 0 / Front-end

Nginx

Back-End

Postgresql
Master



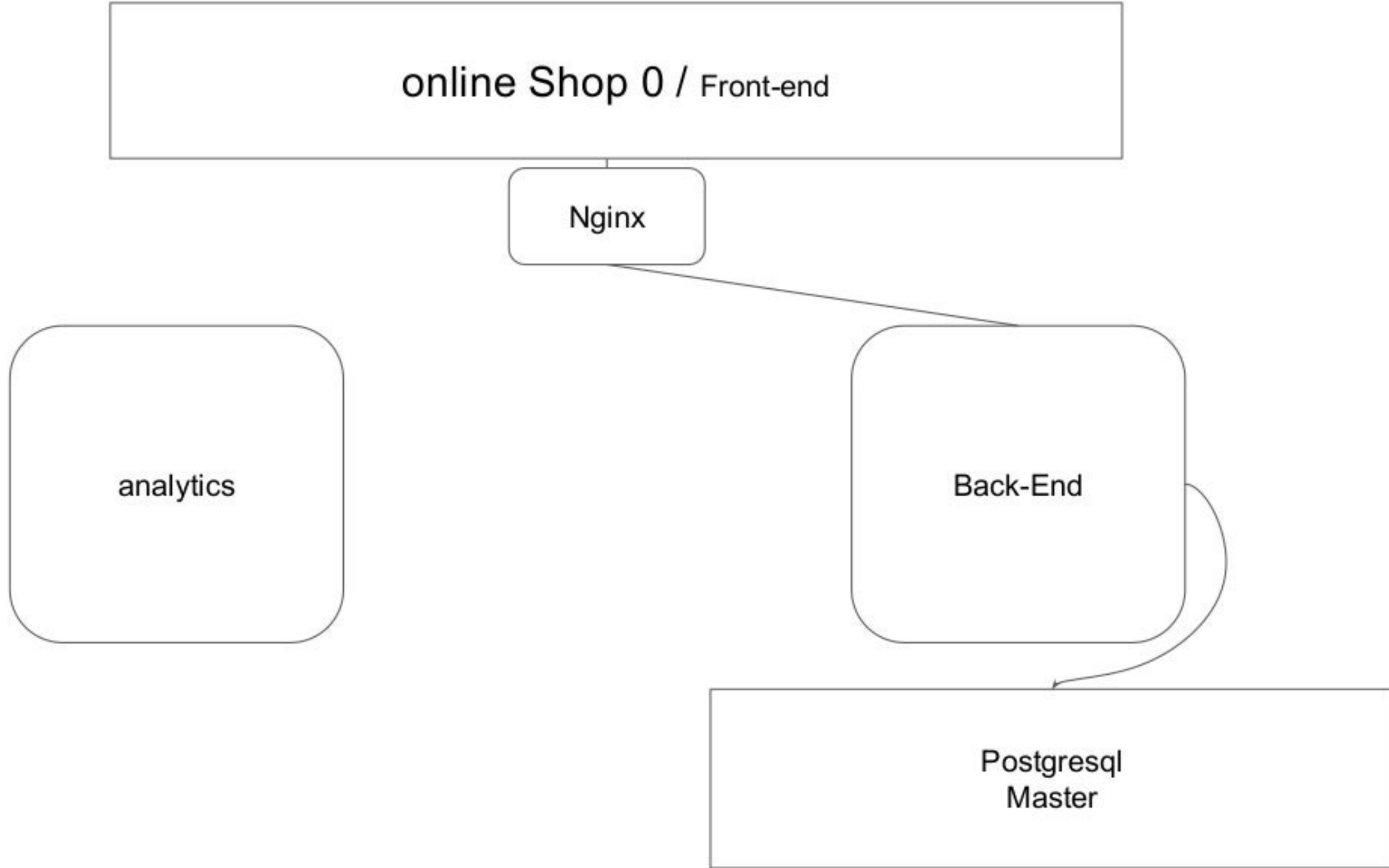
online Shop 0 / Front-end

Nginx

analytics

Back-End

Postgresql
Master



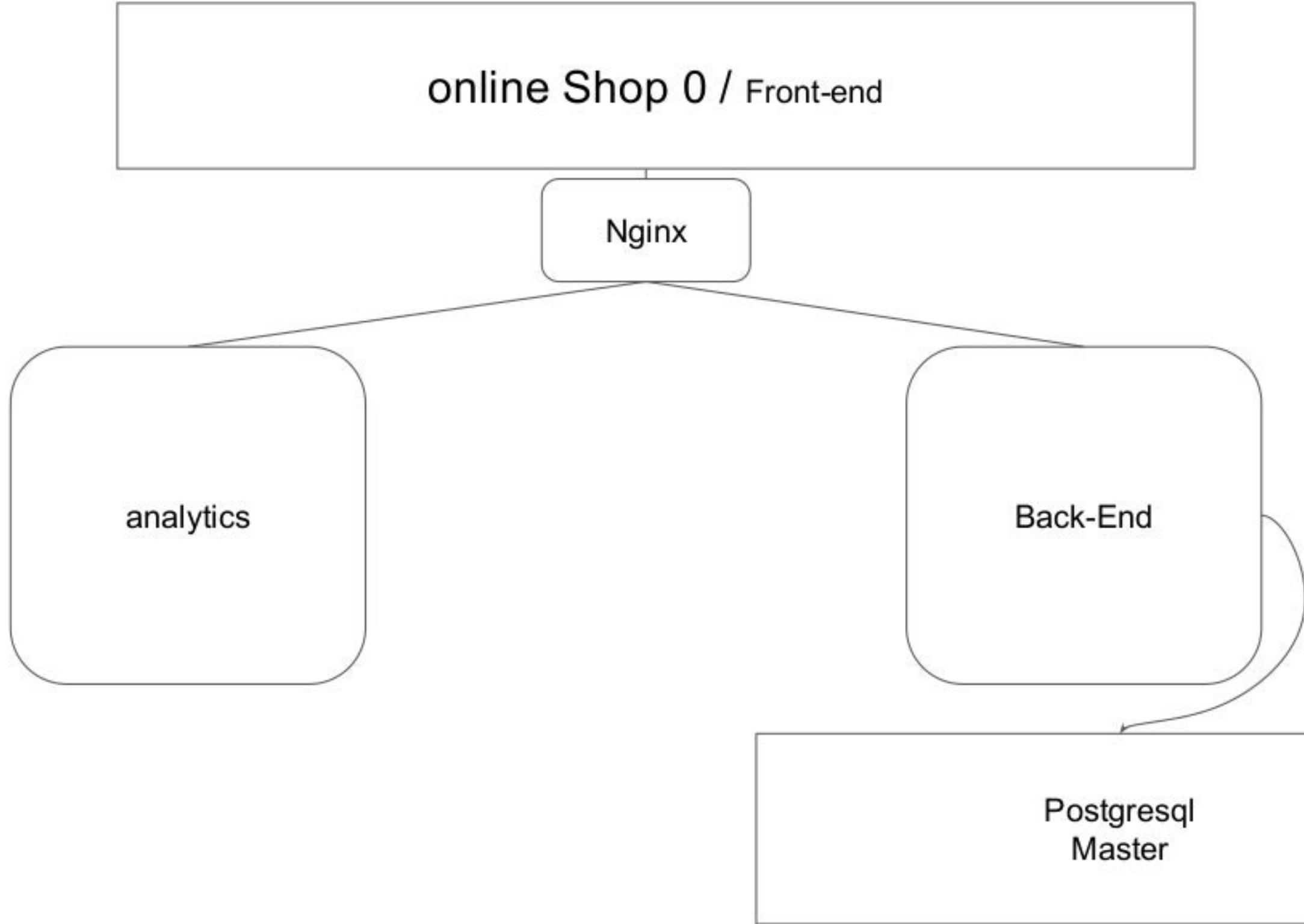
online Shop 0 / Front-end

Nginx

analytics

Back-End

Postgresql
Master



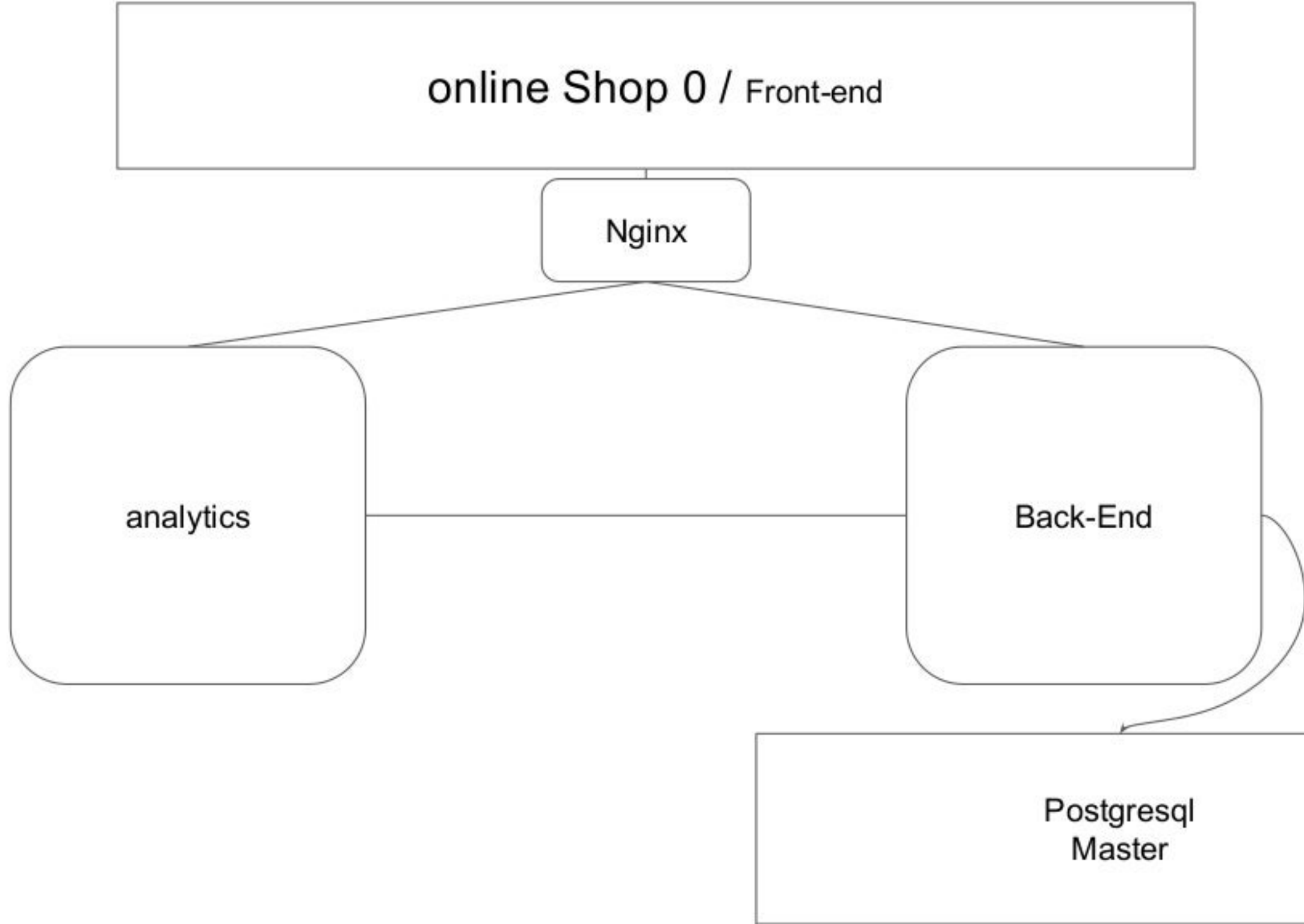
online Shop 0 / Front-end

Nginx

analytics

Back-End

Postgresql
Master



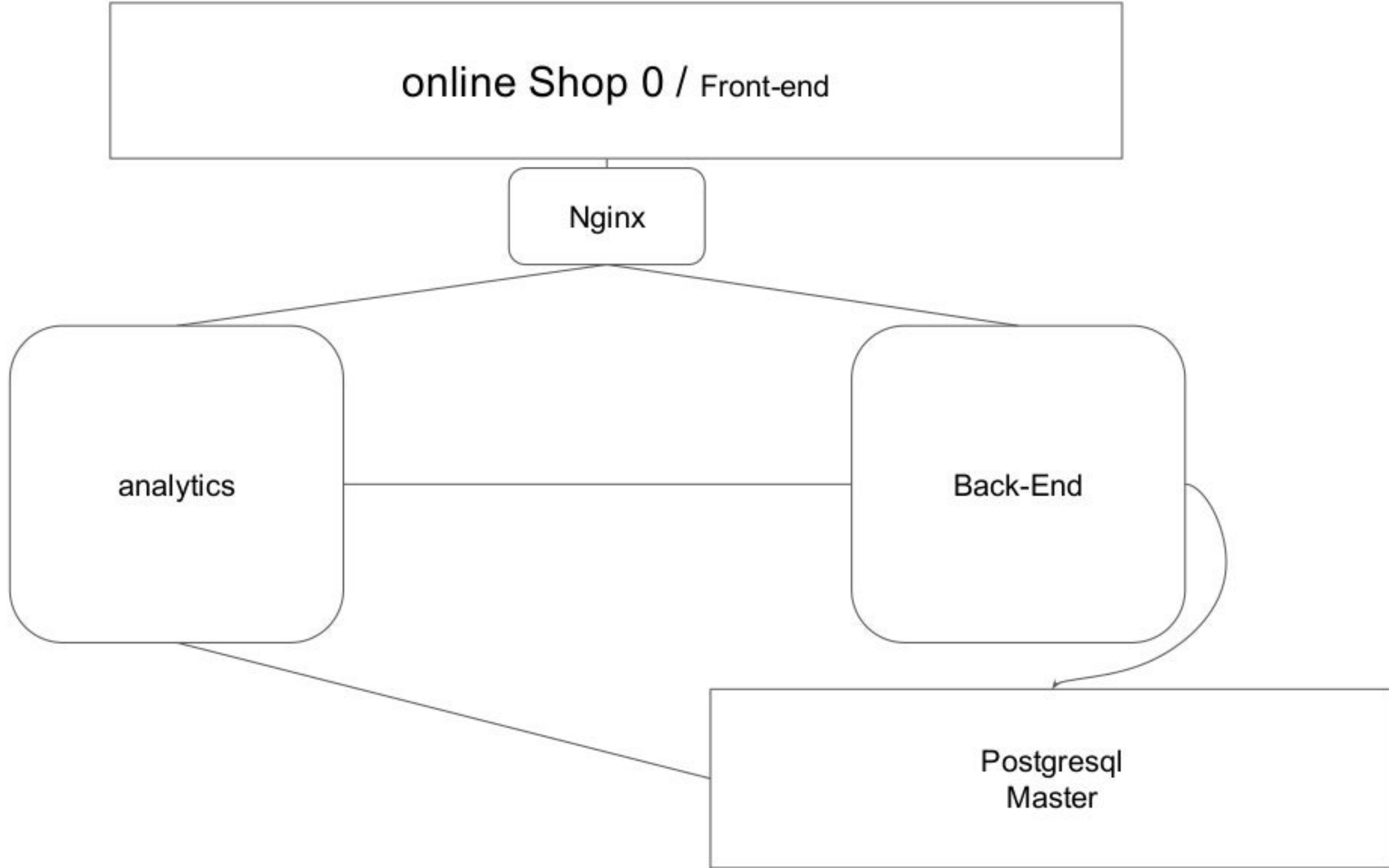
online Shop 0 / Front-end

Nginx

analytics

Back-End

Postgresql
Master



Big Problems:

Big Problems:

- StartUp : 1m - 15m

Big Problems:

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

Big Problems:

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

Big Problems:

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

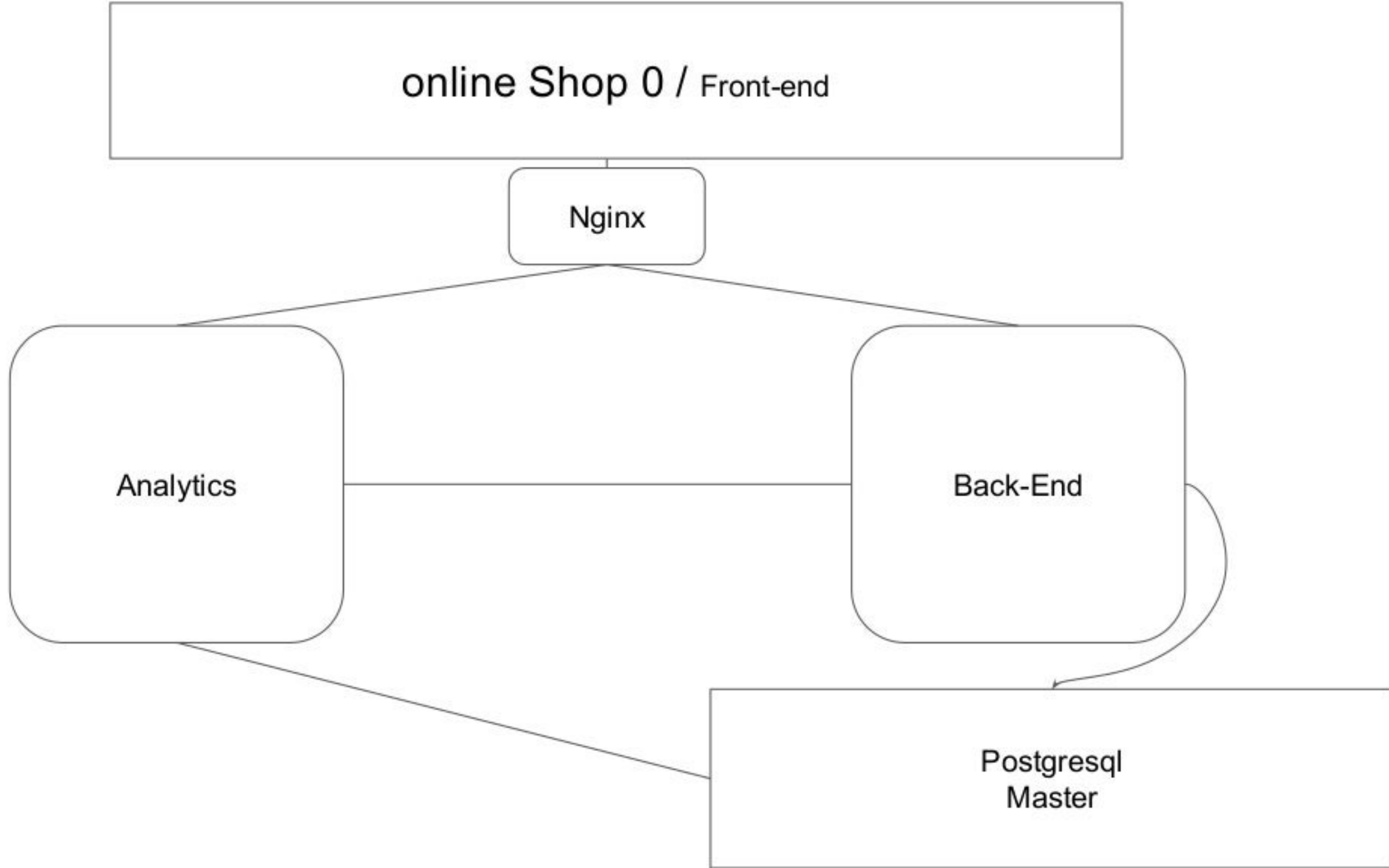
online Shop 0 / Front-end

Nginx

Analytics

Back-End

Postgresql
Master



online Shop 0 / Front-end

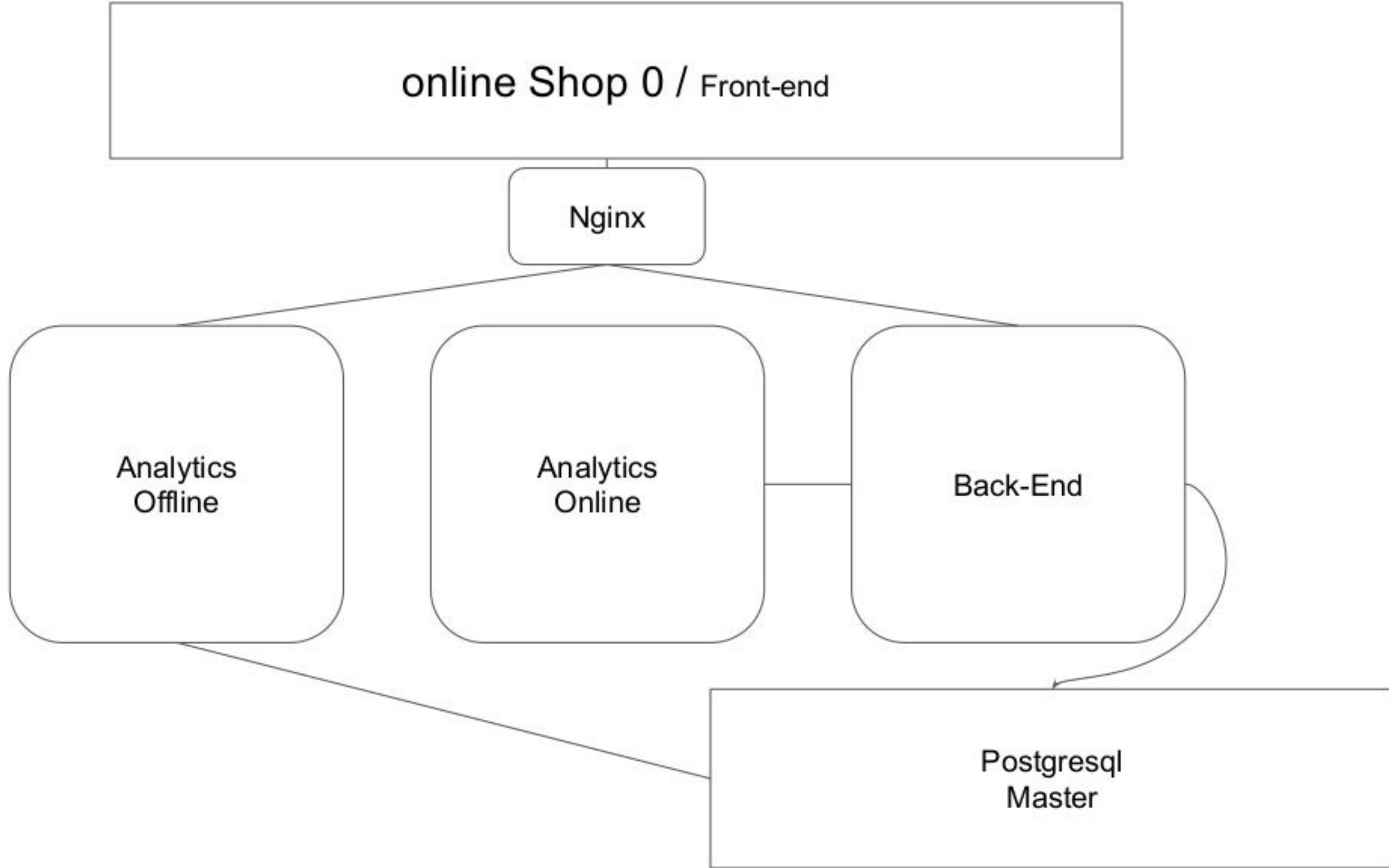
Nginx

Analytics
Offline

Analytics
Online

Back-End

Postgresql
Master



online Shop 0 / Front-end

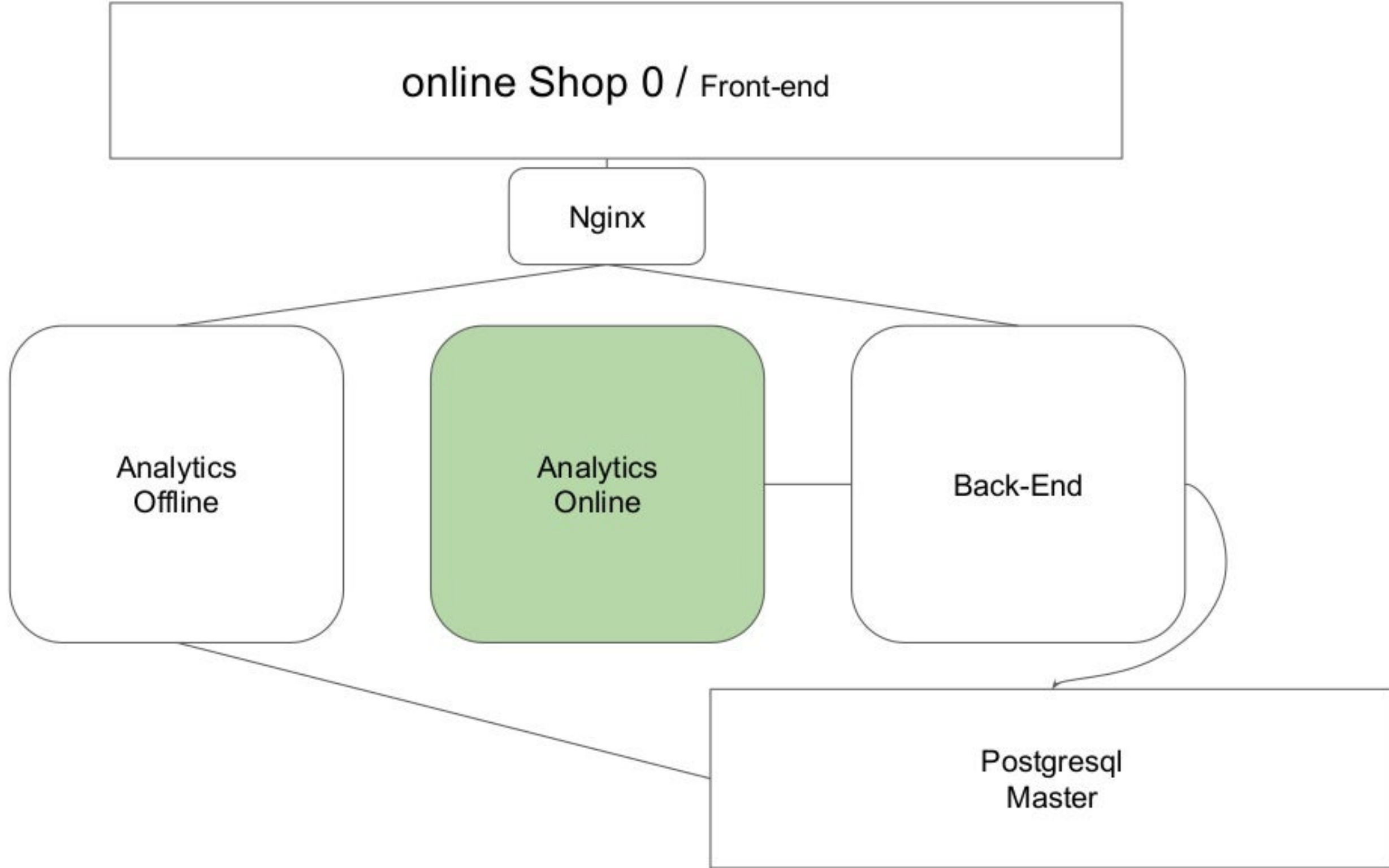
Nginx

Analytics
Offline

Analytics
Online

Back-End

Postgresql
Master



online Shop 0 / Front-end

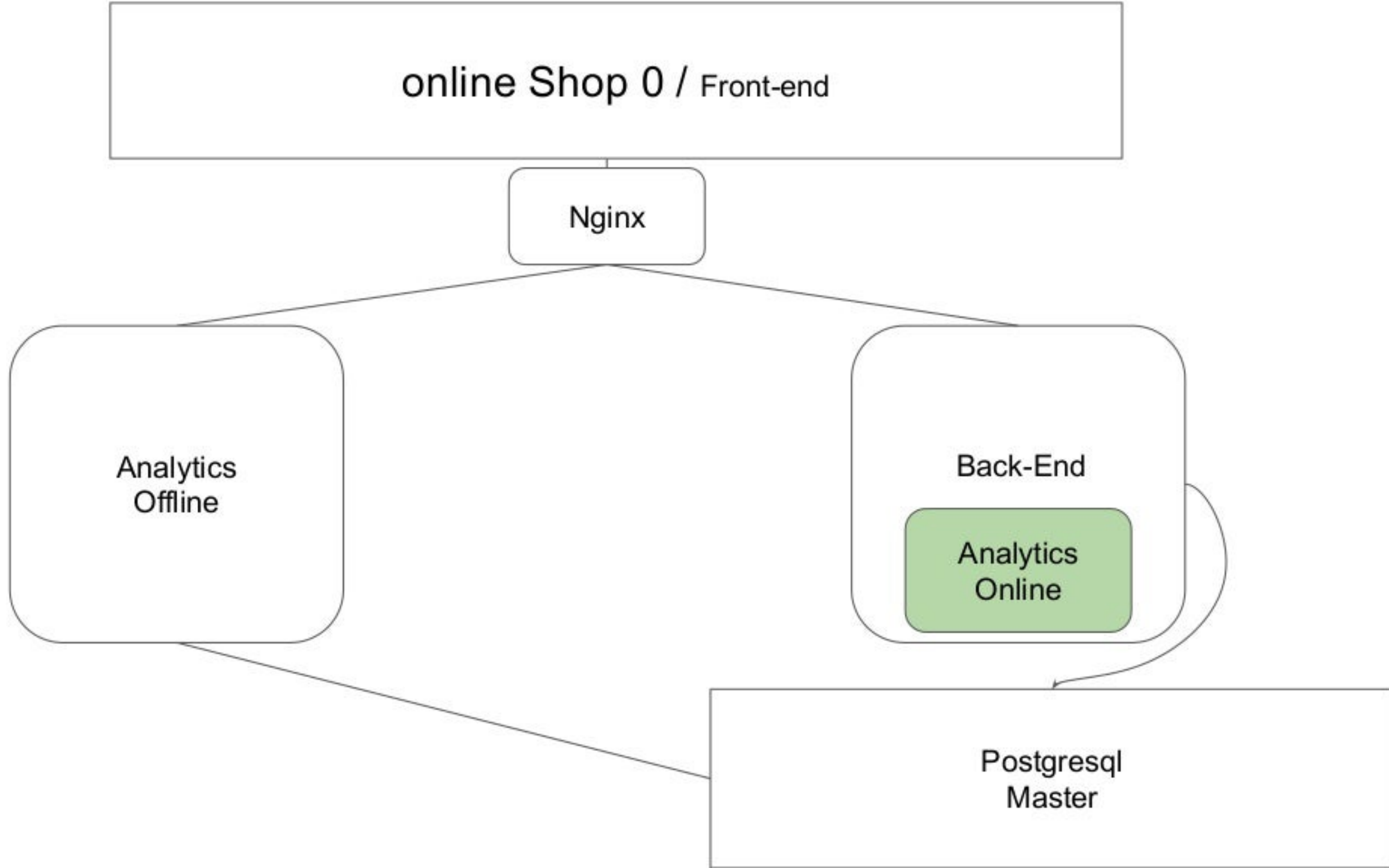
Nginx

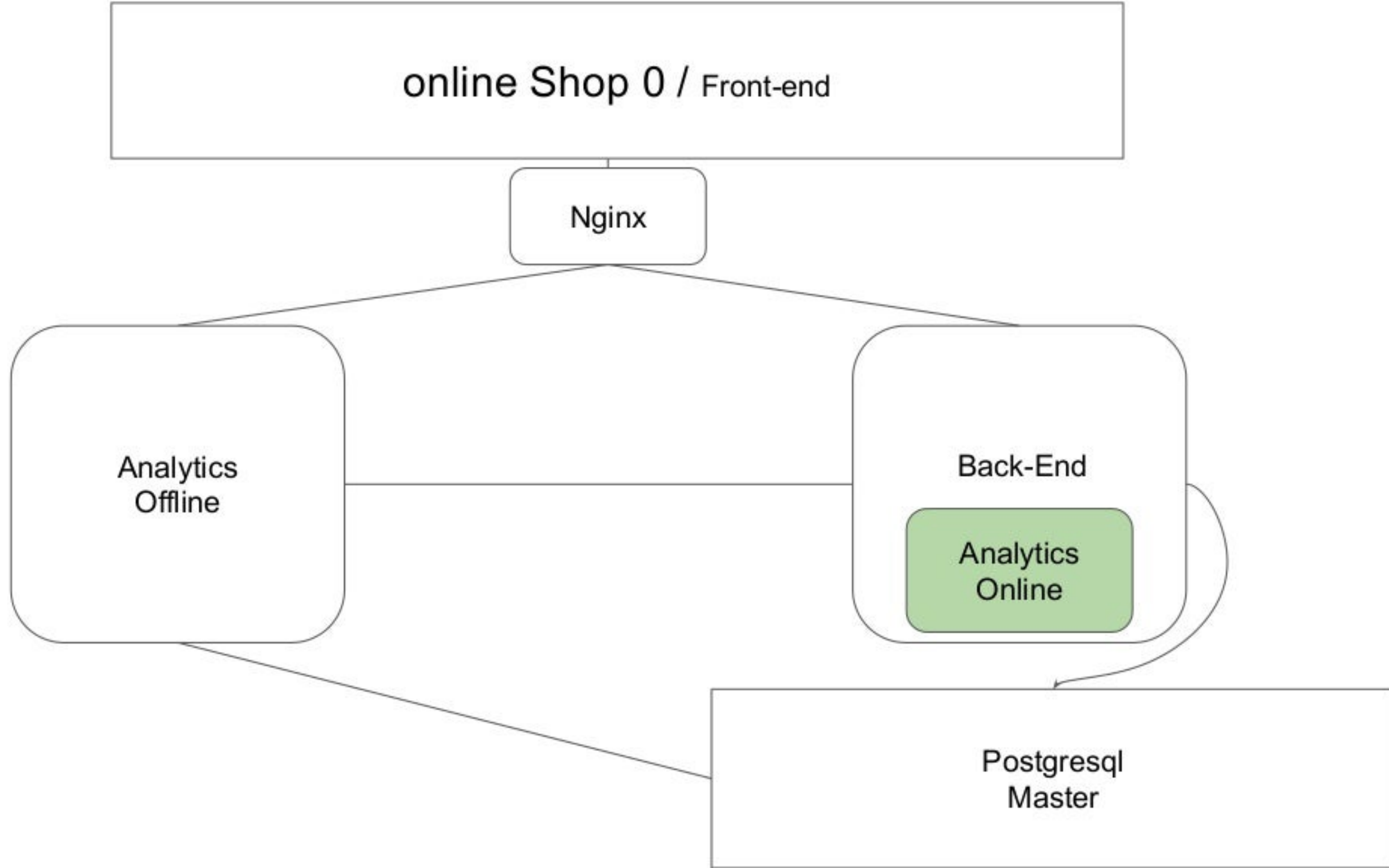
Analytics
Offline

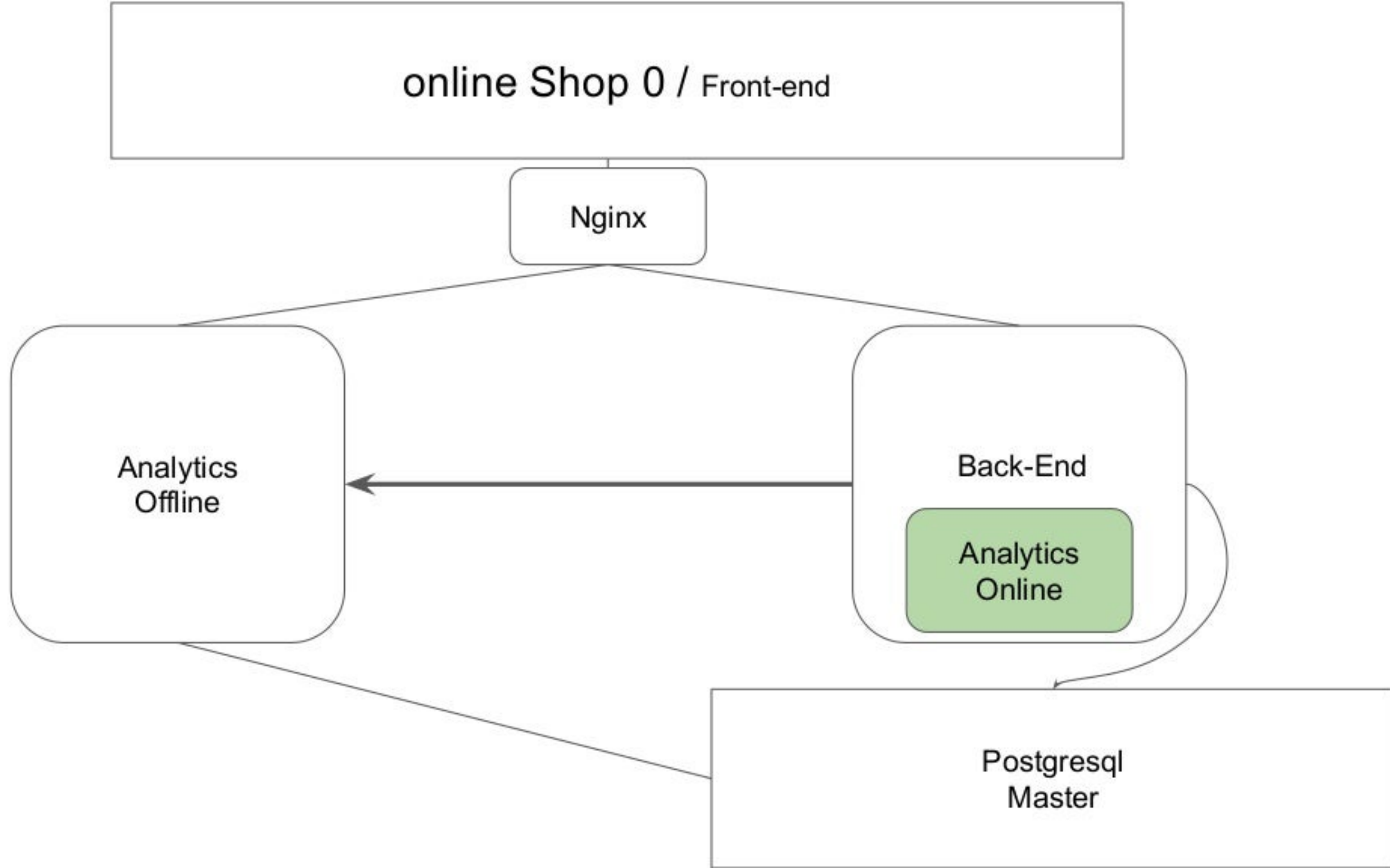
Back-End

Analytics
Online

Postgresql
Master







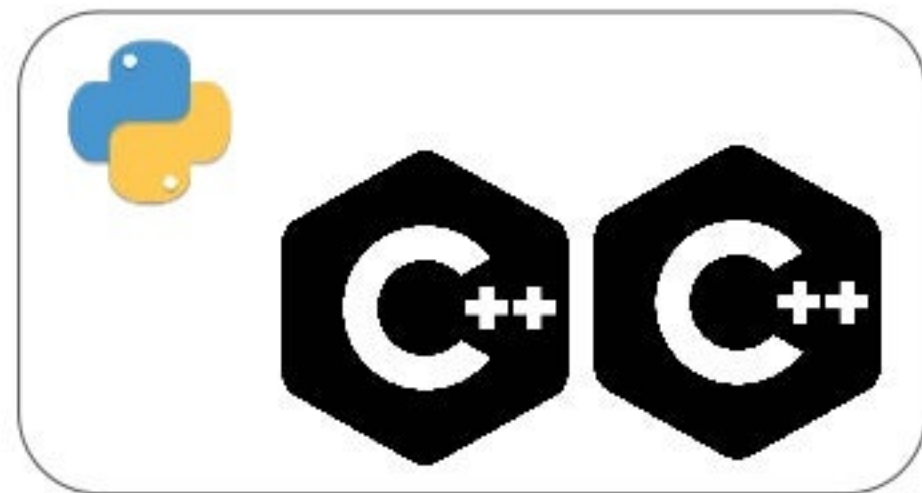
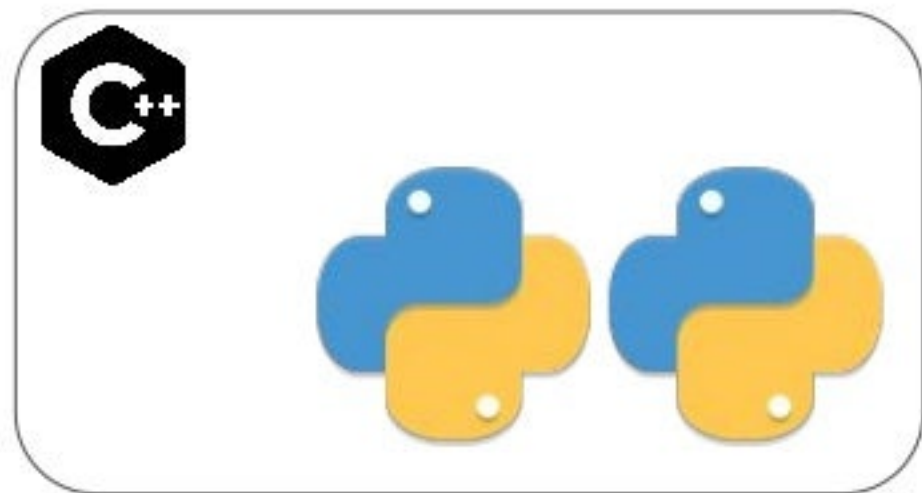
Big Problems:

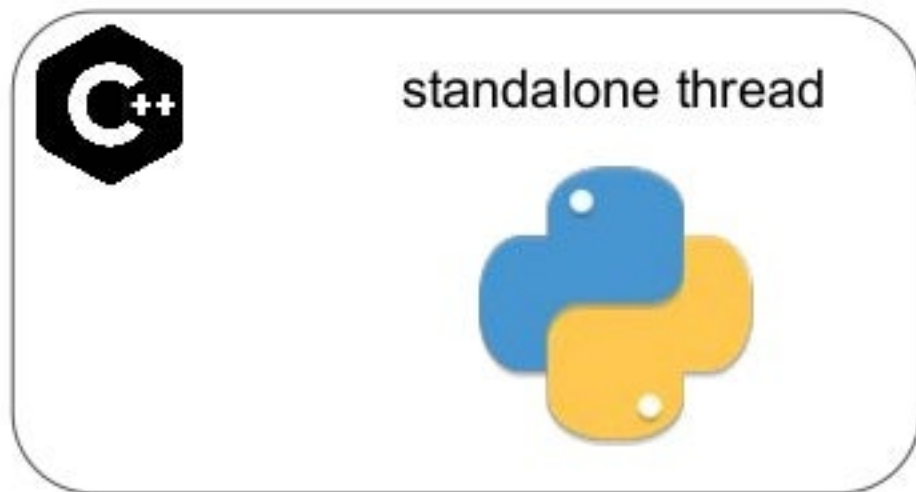
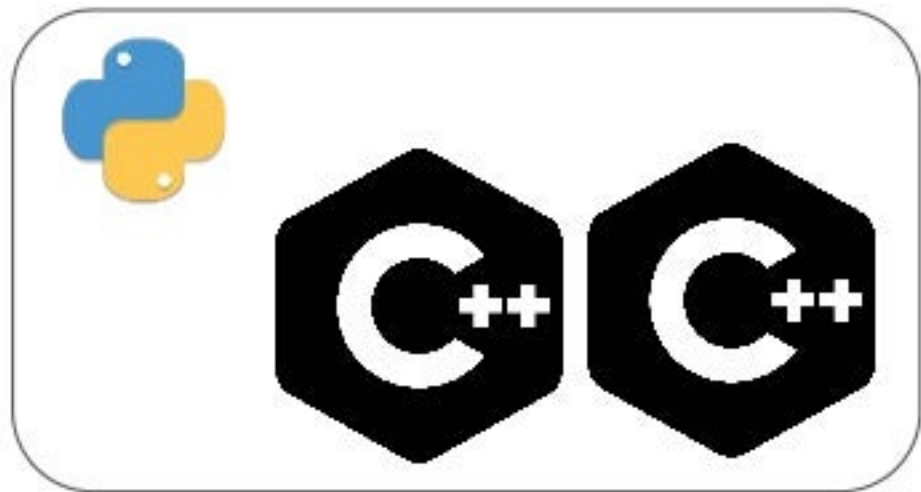
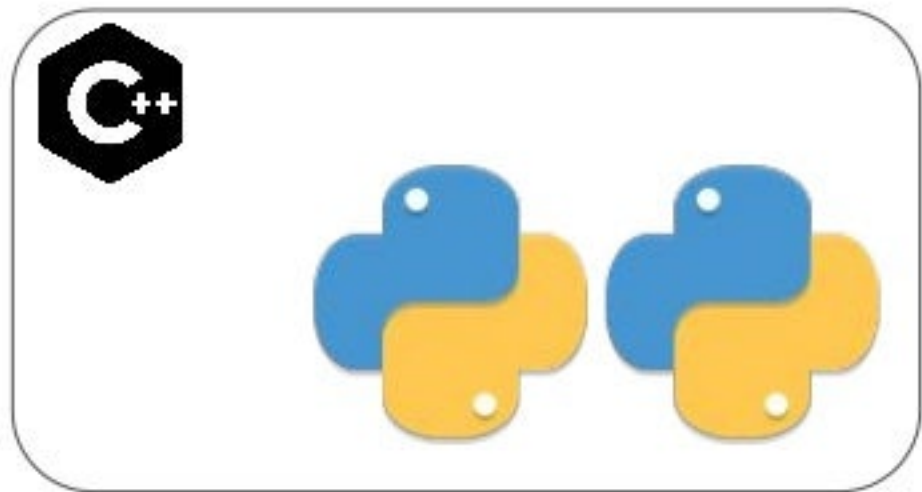
- StartUp : 1m - 15m
- Response latency : 1s - 1m
- Python
- Number of machines: 100

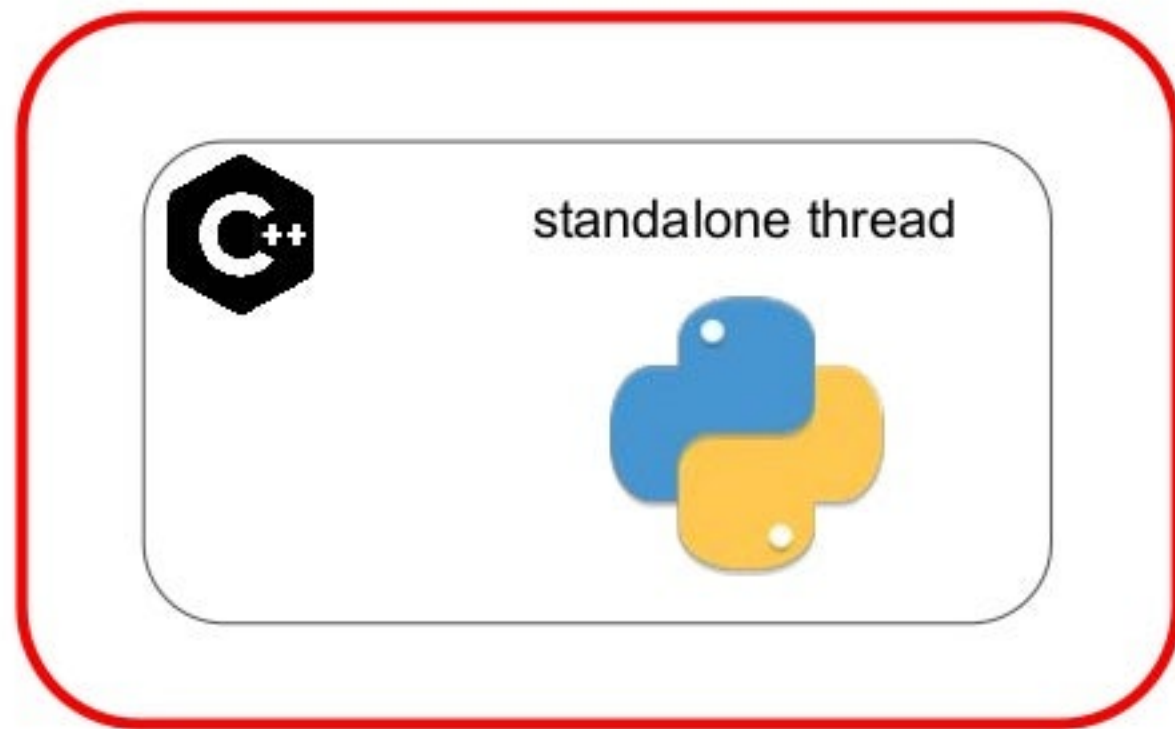
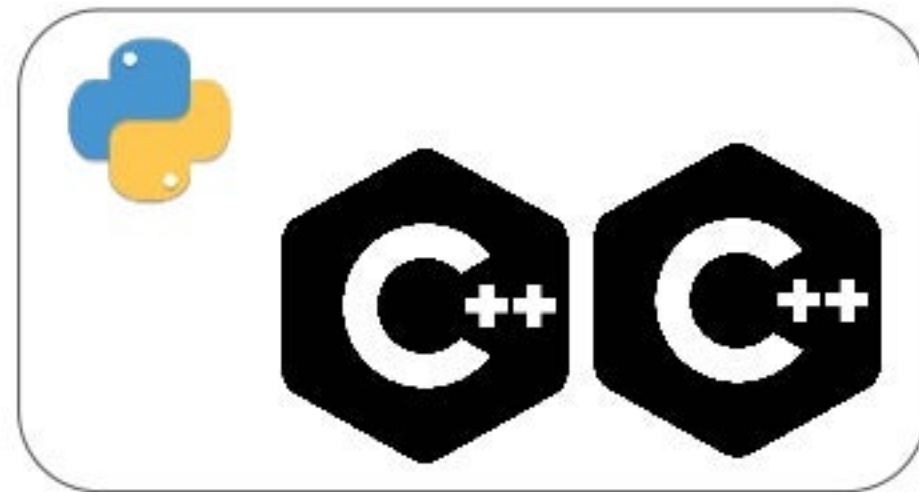
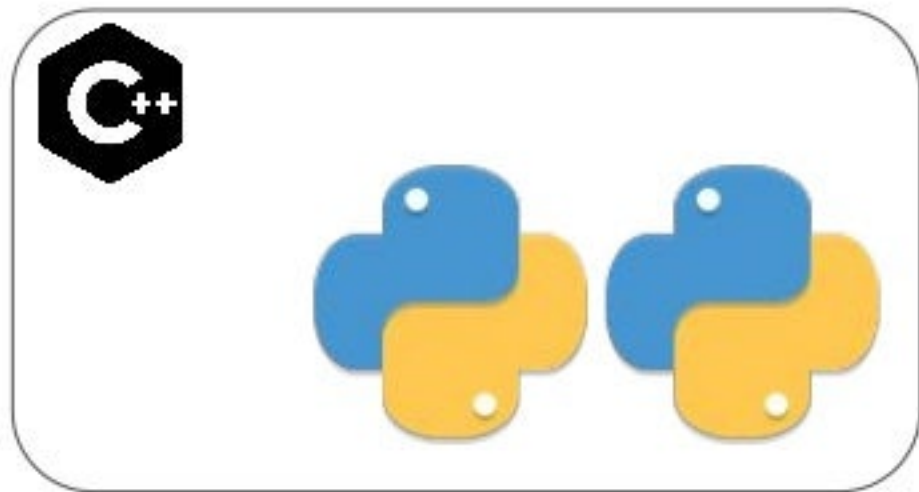
Big Problems:

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

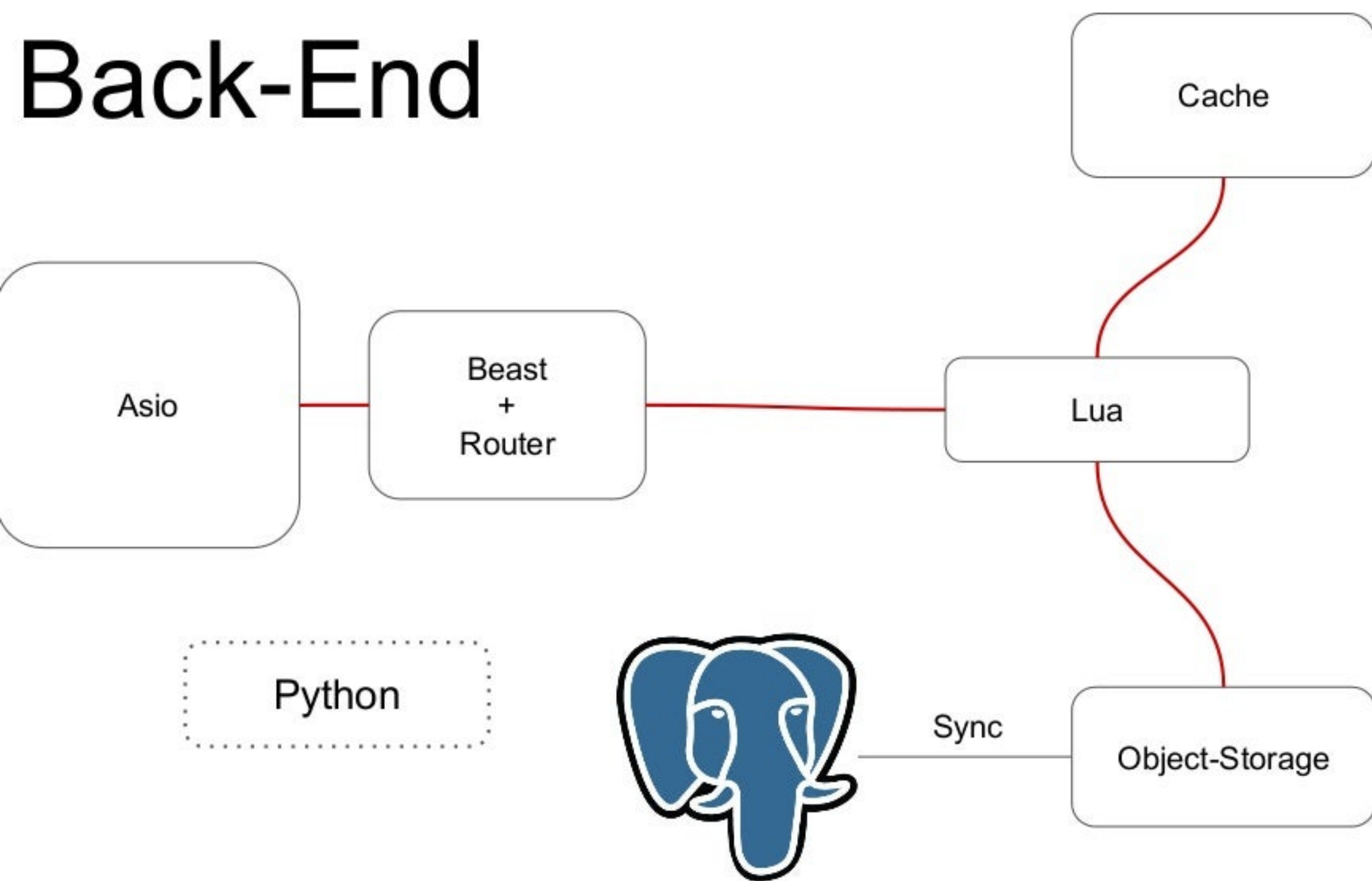


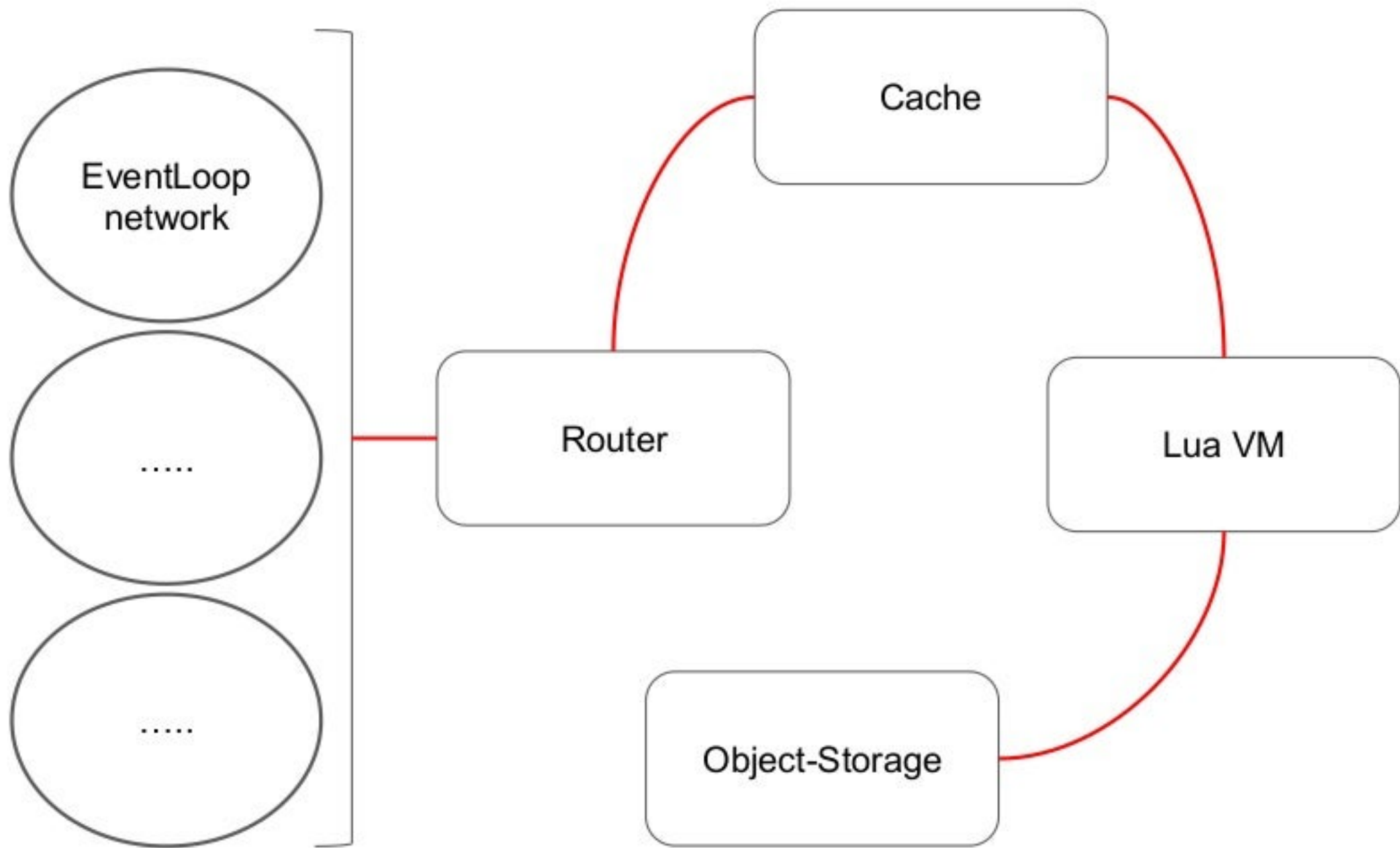


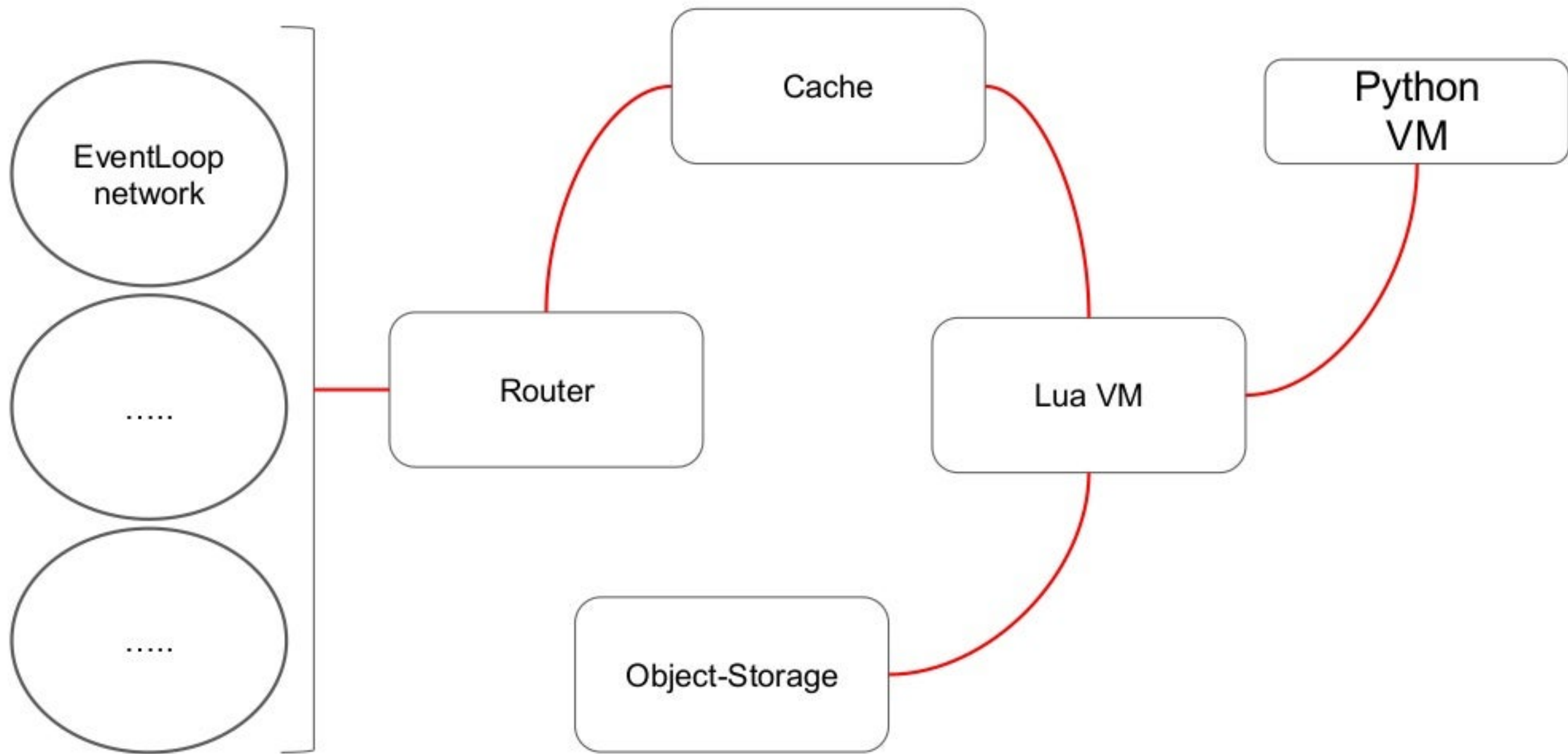


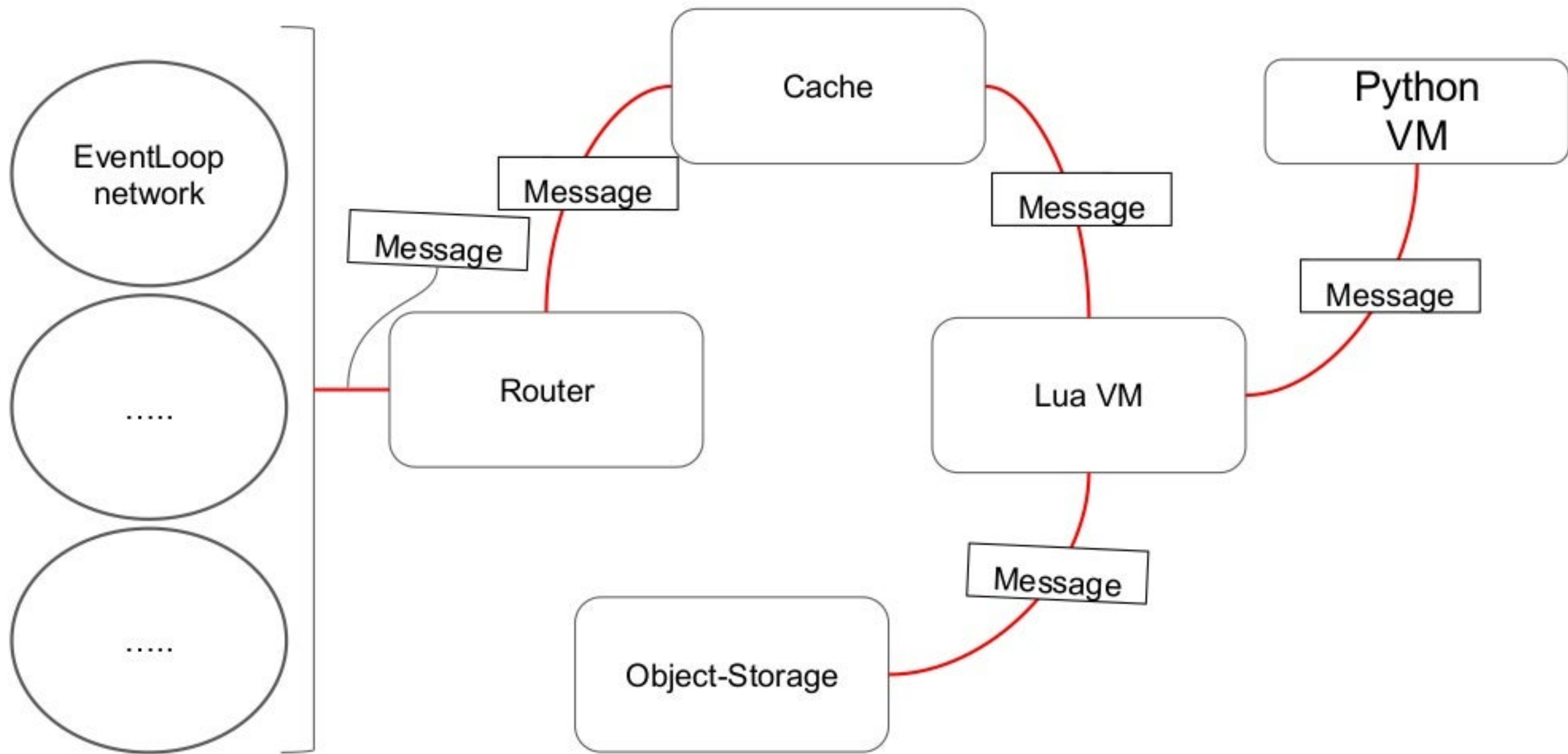


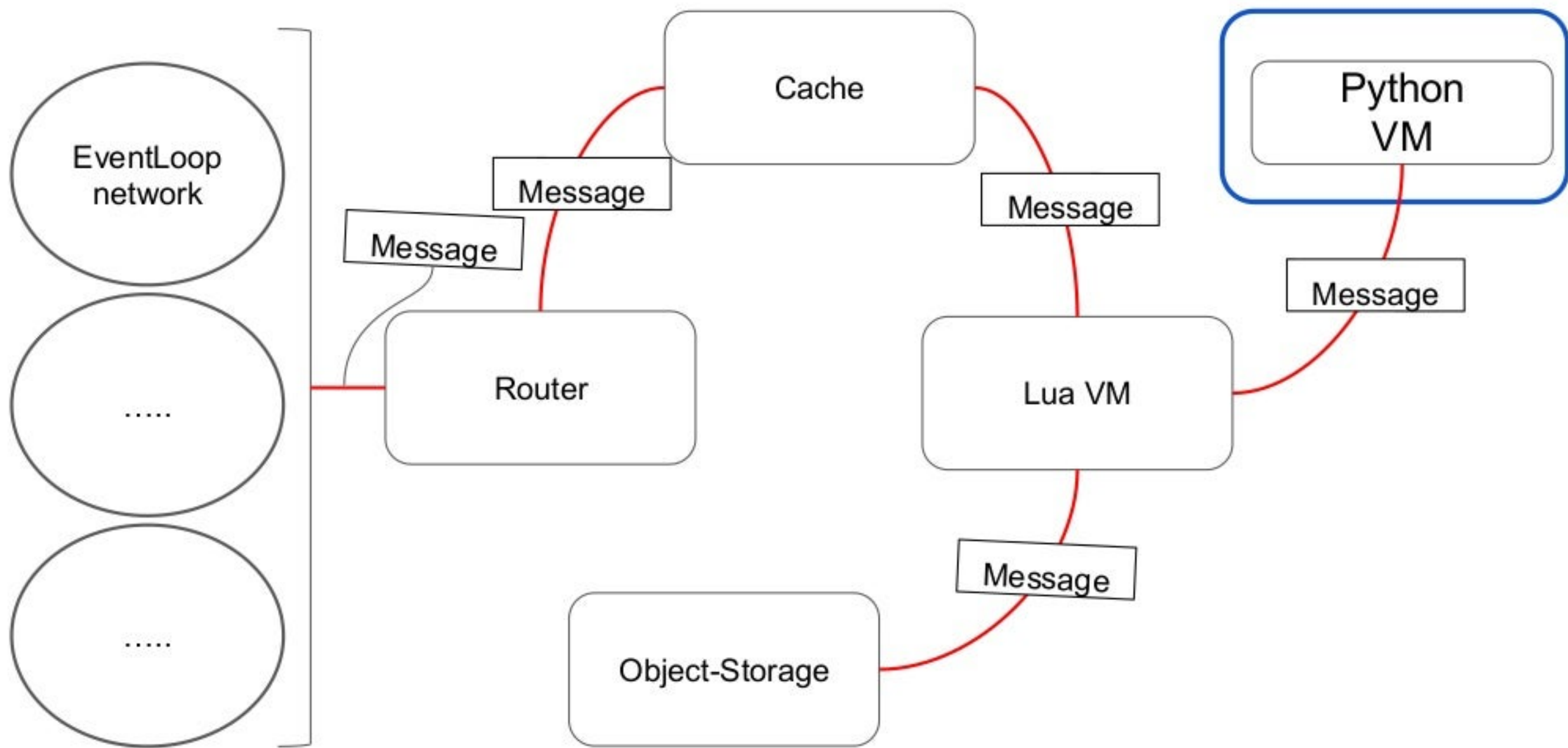
Back-End











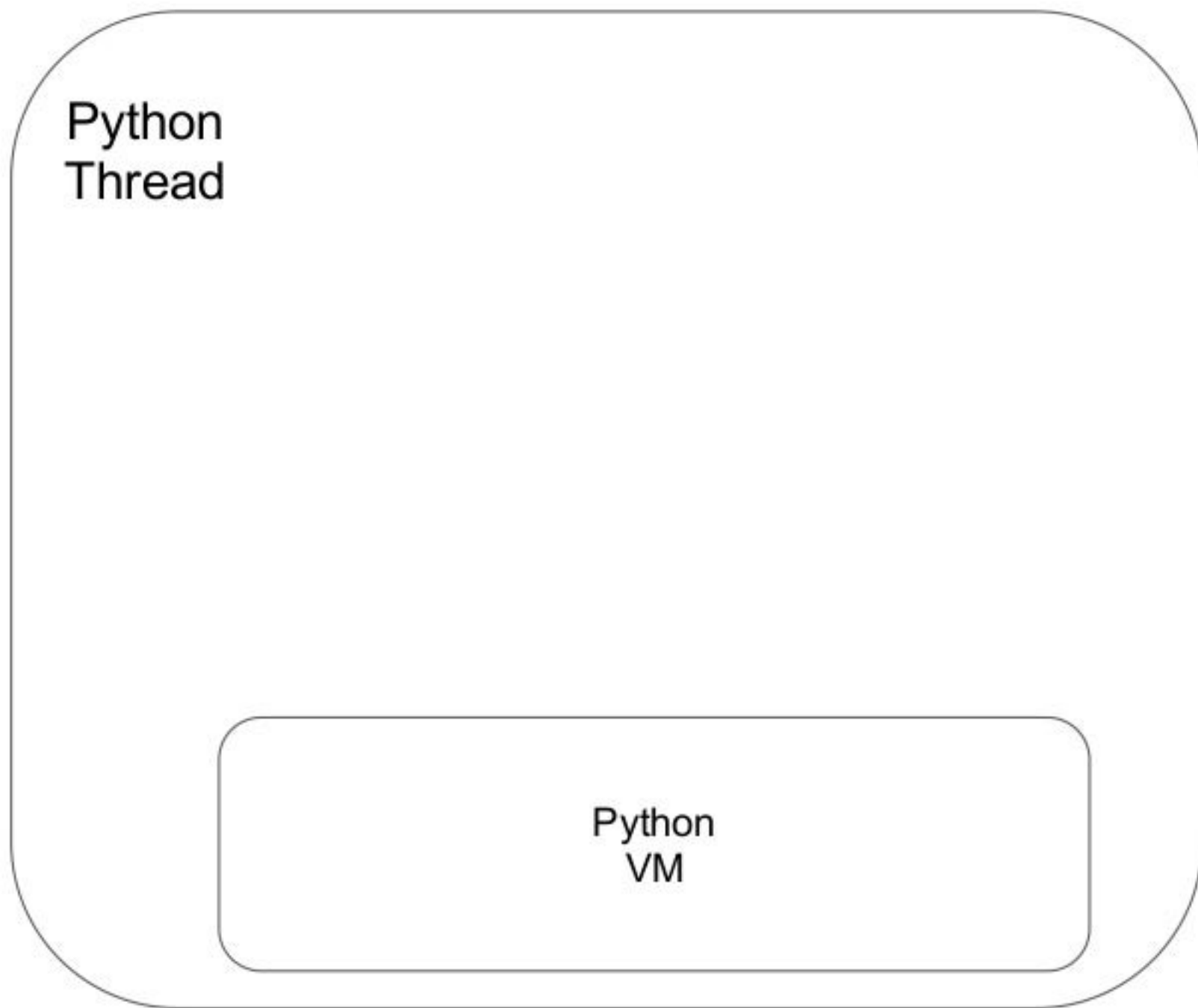
Python
VM

Python VM

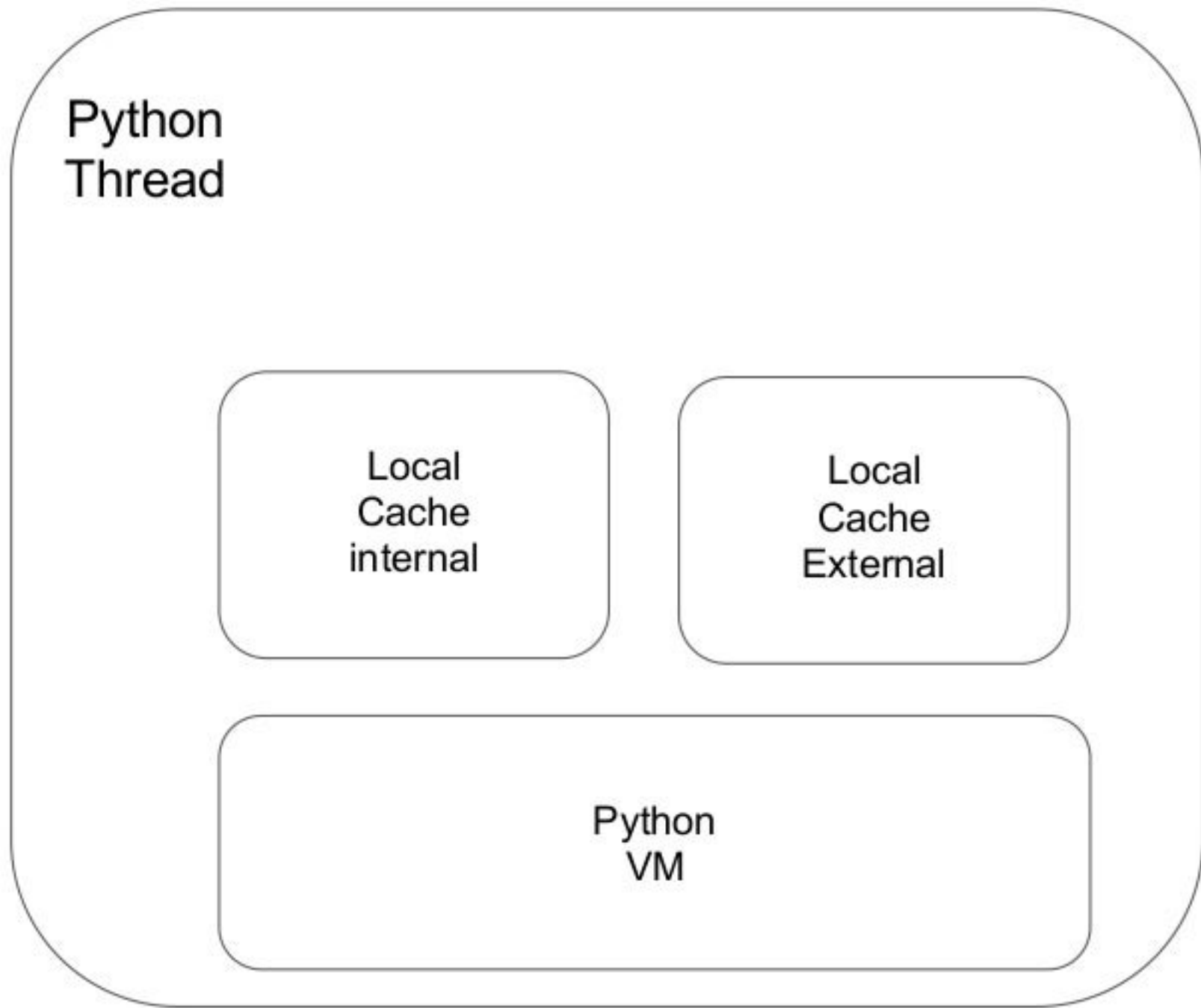
Python
Thread

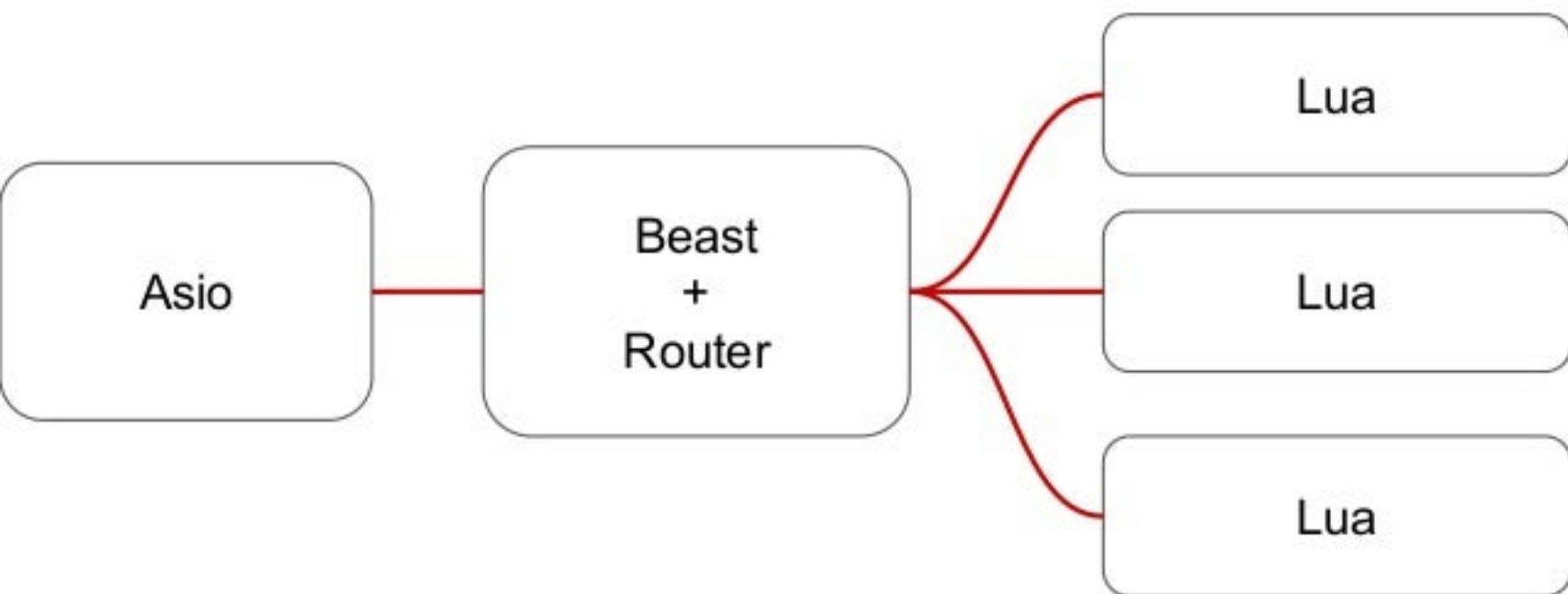
A diagram illustrating the relationship between a Python VM and a Python Thread. On the left, the text "Python VM" is displayed. To its right is a large, light-blue rounded rectangle representing the VM. Inside the top-left corner of this rectangle, the text "Python Thread" is written, indicating that a thread exists within the virtual machine environment.

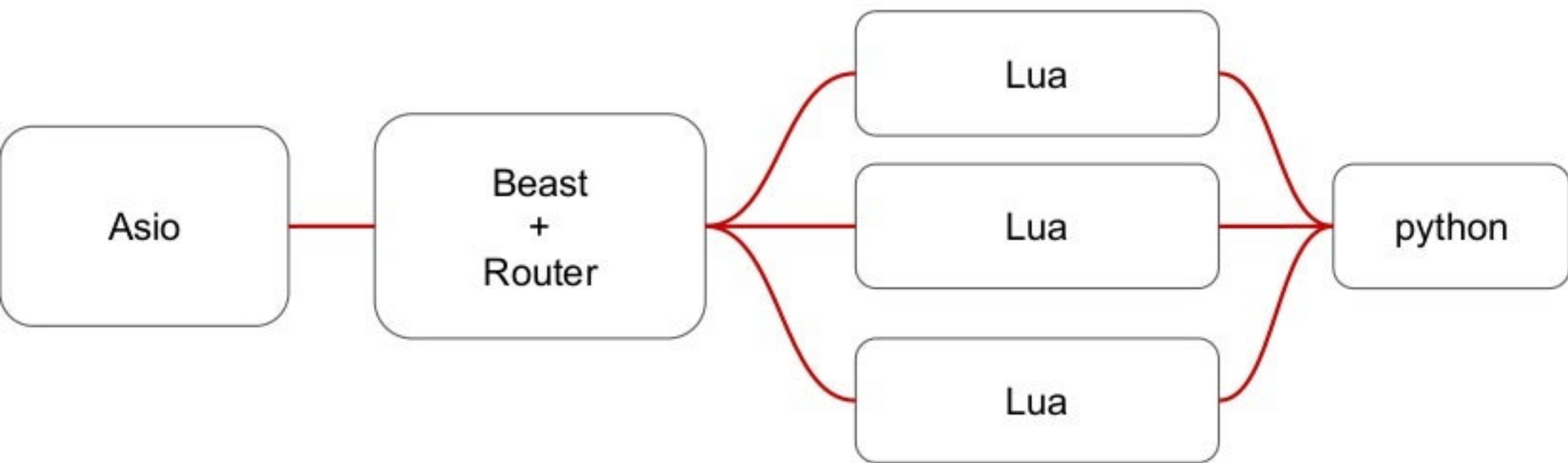
Python VM

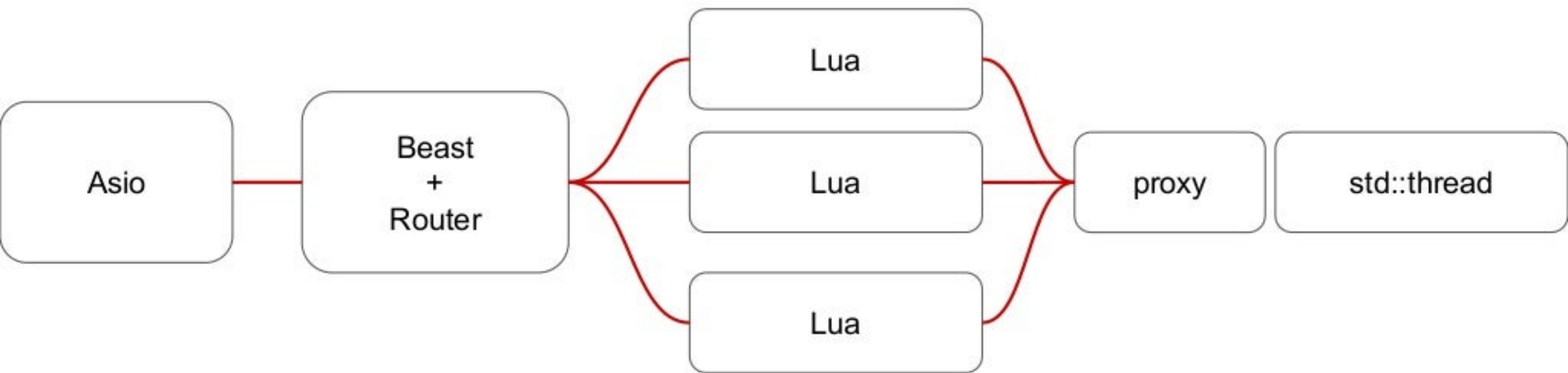


Python VM









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



```
#!/usr/bin/env python3
#...
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:
```

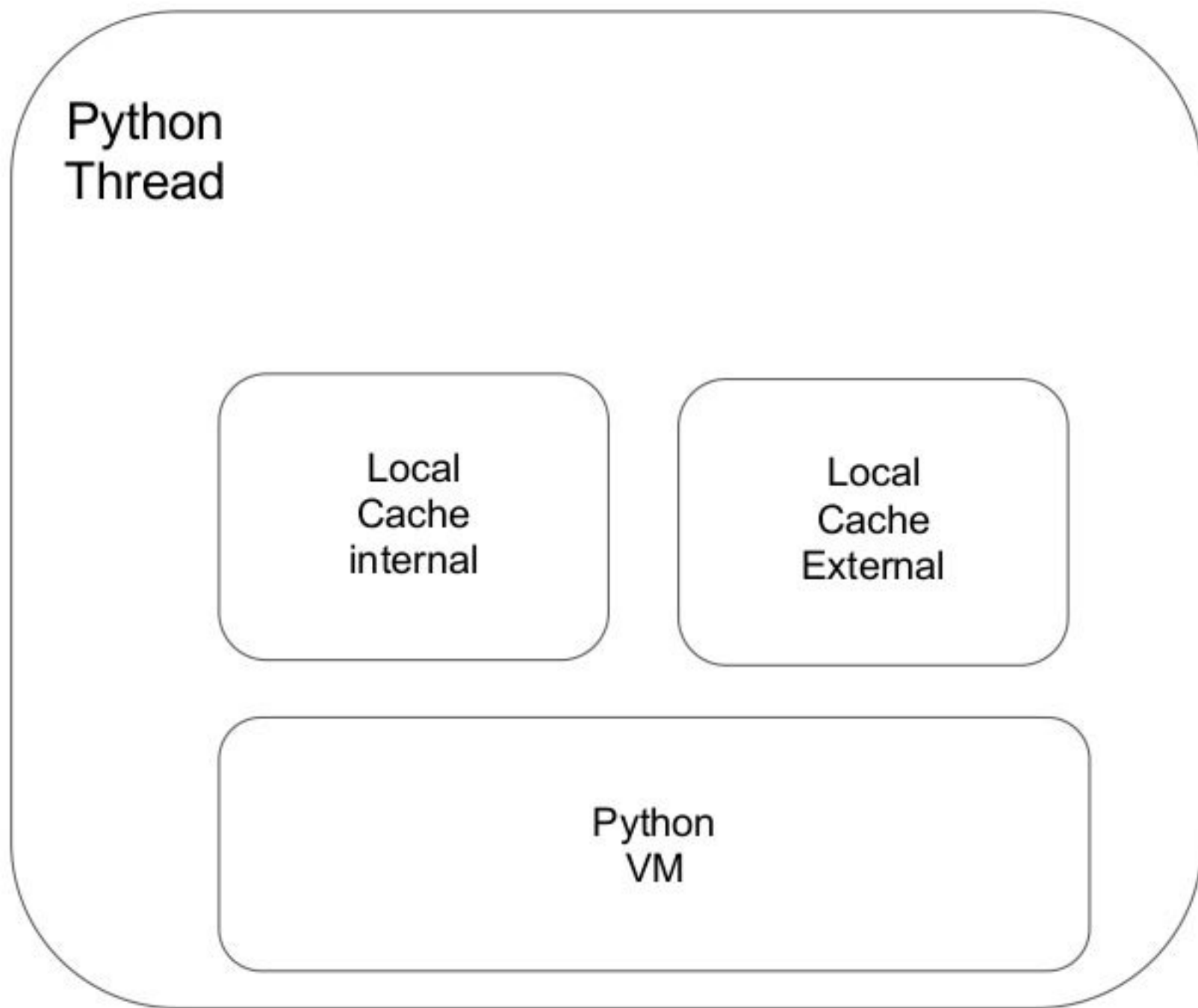
```
    data = file.readline()
```

```
    data_storage.update(transform(data))
```

```
#...
```

```
return app
```

Python VM



Python VM

`unordered_map<string,File>`

Python
Thread

Local
Cache
internal

Local
Cache
External

Python
VM

Python VM

`unordered_map<string,File>`

File

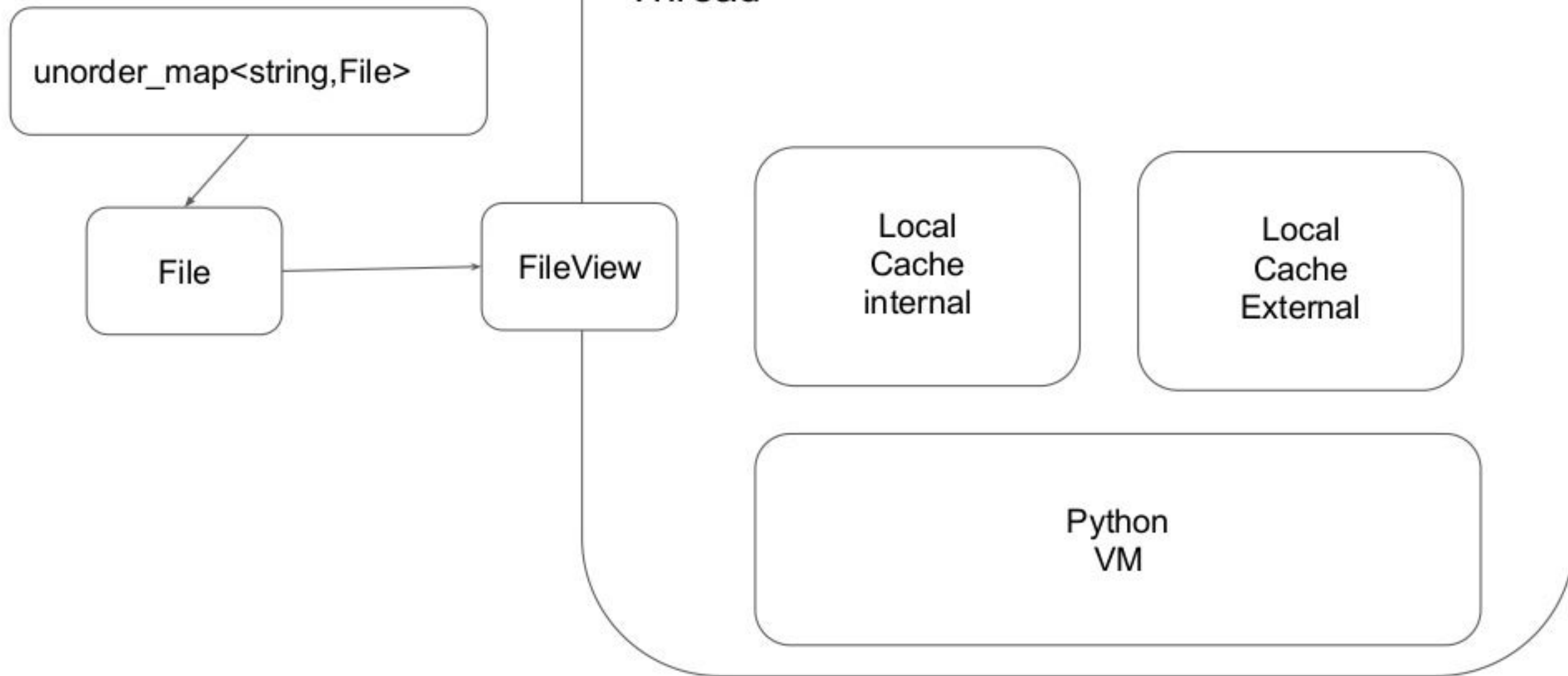
Python
Thread

Local
Cache
internal

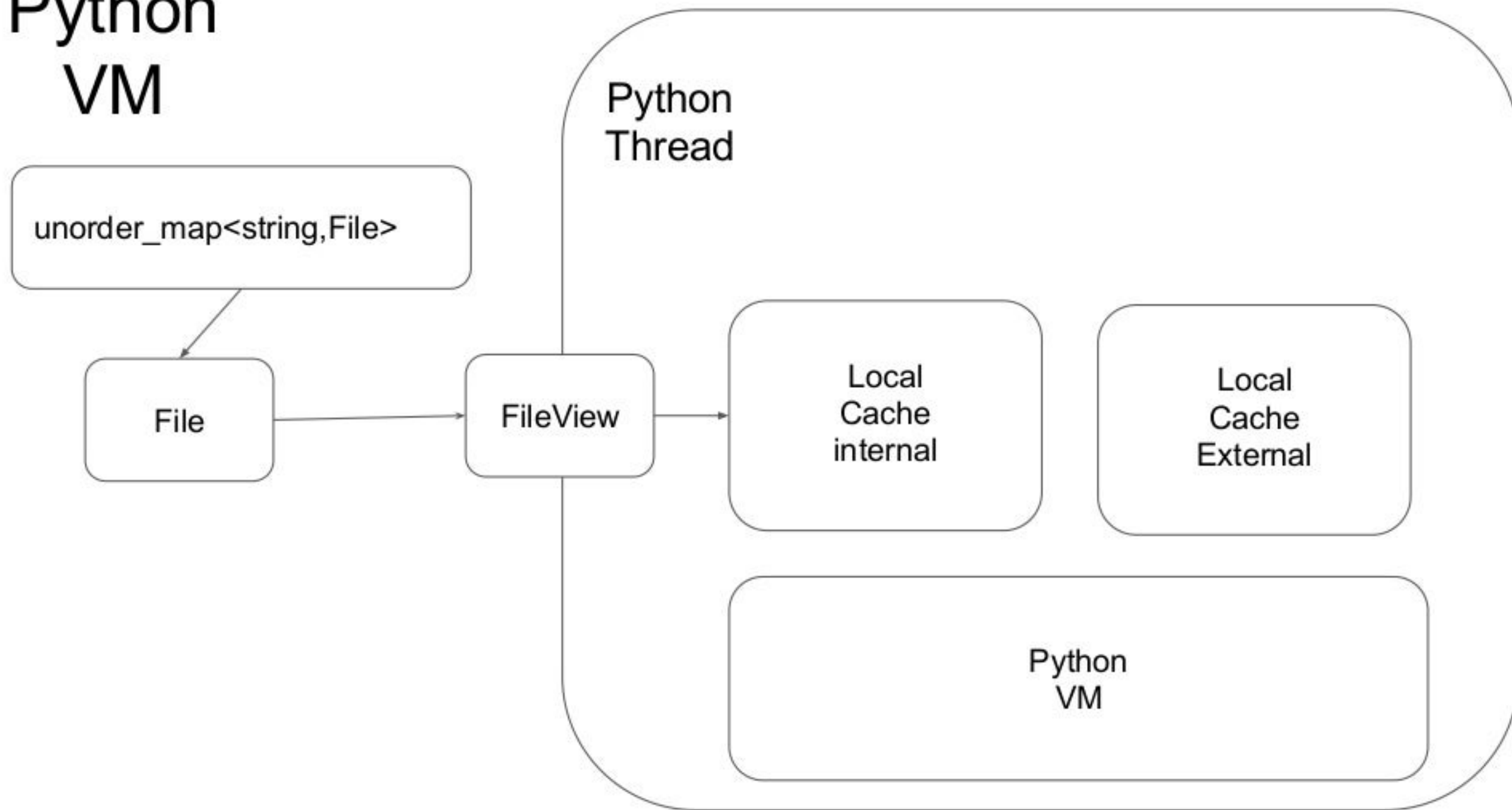
Local
Cache
External

Python
VM

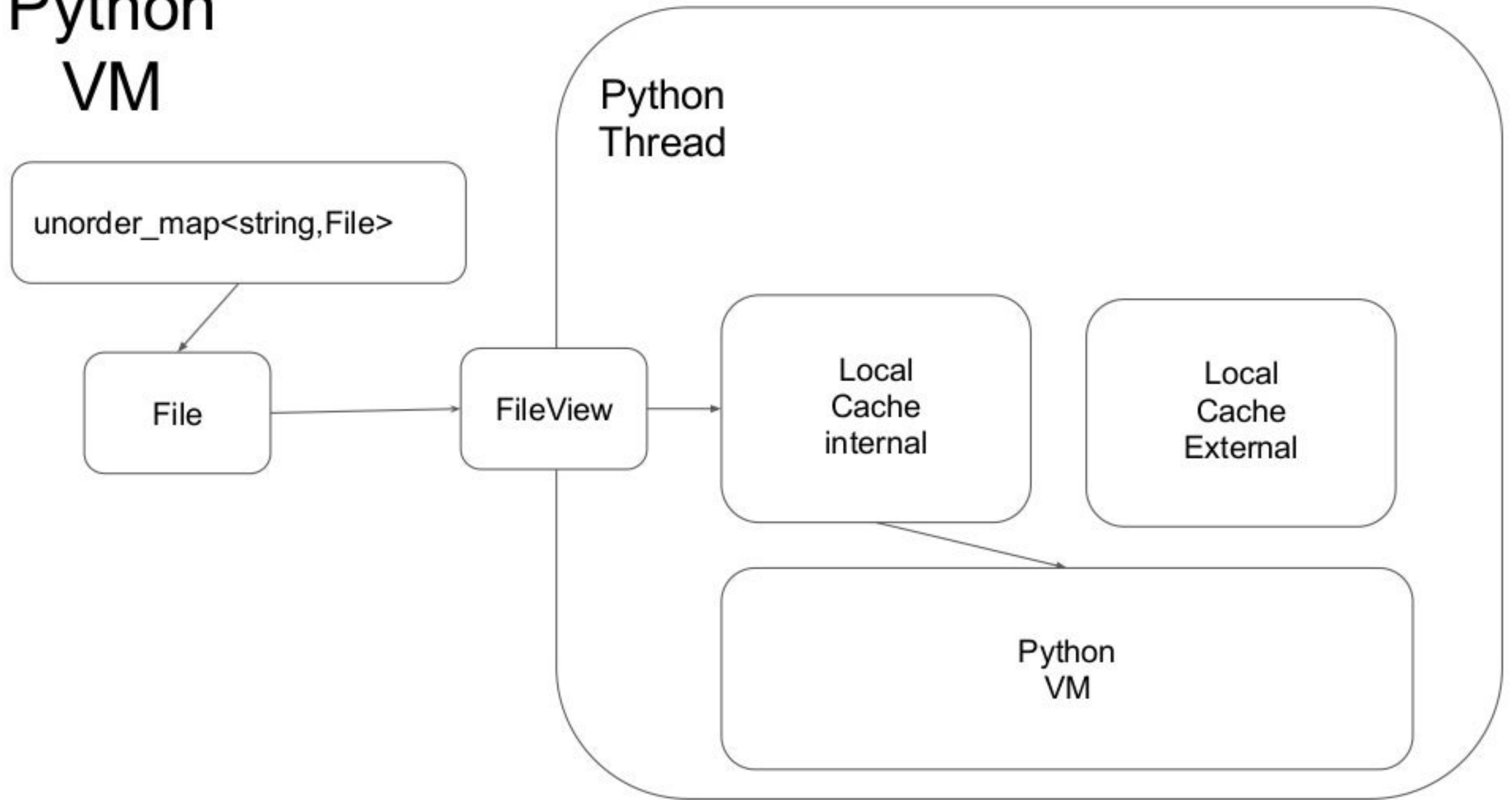
Python VM



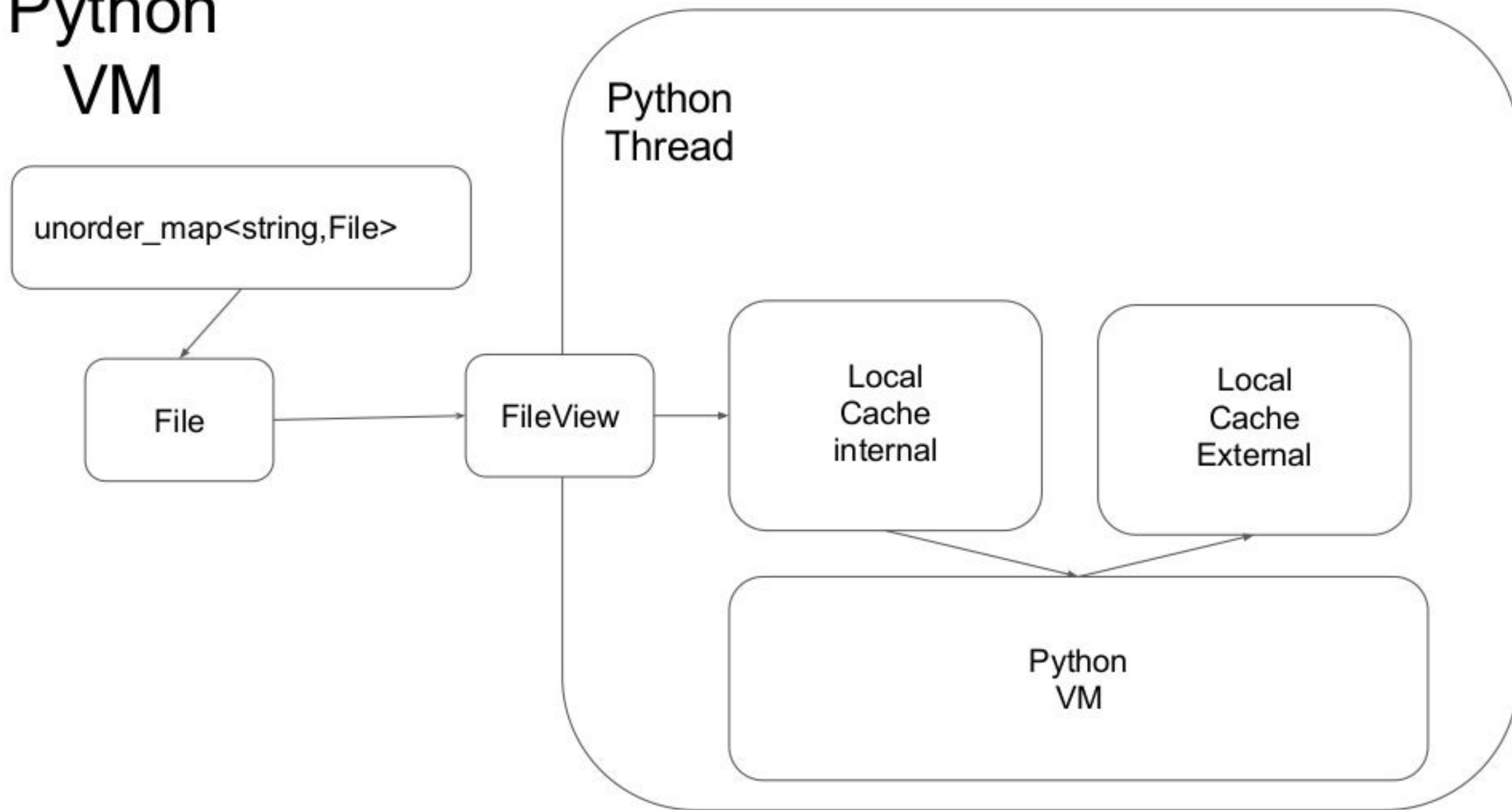
Python VM



Python VM



Python VM



#...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
```

```
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) {  
        ///...  
    }  
);
```



```
exuctor = 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);  
        ///...  
    }  
);
```

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

spdlog
Rapidjson
Boost.asio
Boost.beast
pybind11
yaml-cpp
sol
cxxopts
conan



Borgardt Alexander

aa.borgardt@yandex.ru

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

Borgardt Alexander

✉ **aa.borgardt@yandex.ru**