

Яндекс

Яндекс

# Введение в uWSGI

Игорь Стариков

Что такое uWSGI



uWSGI — решение для веб-приложений,  
предлагающее широчайший набор  
инструментов.

uWSGI в Яндексе





- Приложение Яндекс
- Такси
- Директ
- Диск
- КиноПоиск
- Проекты Издательства Яндекса

- Маркет
- Коллекции
- Район
- Поддержка пользователей
- и пр.

# Инструментарий



# Обзор

- **Рабочие, мулы и ведущий процесс**
- Кэширование
- Блокировки
- Сигналы и оповещения
- Мониторинг
- Протоколирование
- RPC
- Планировщик и очередь
- Управление процессами
- Адаптация к нагрузке
- Маршрутизация, балансировка
- Веб-сокеты, поддержка асинхронного кода и пр.

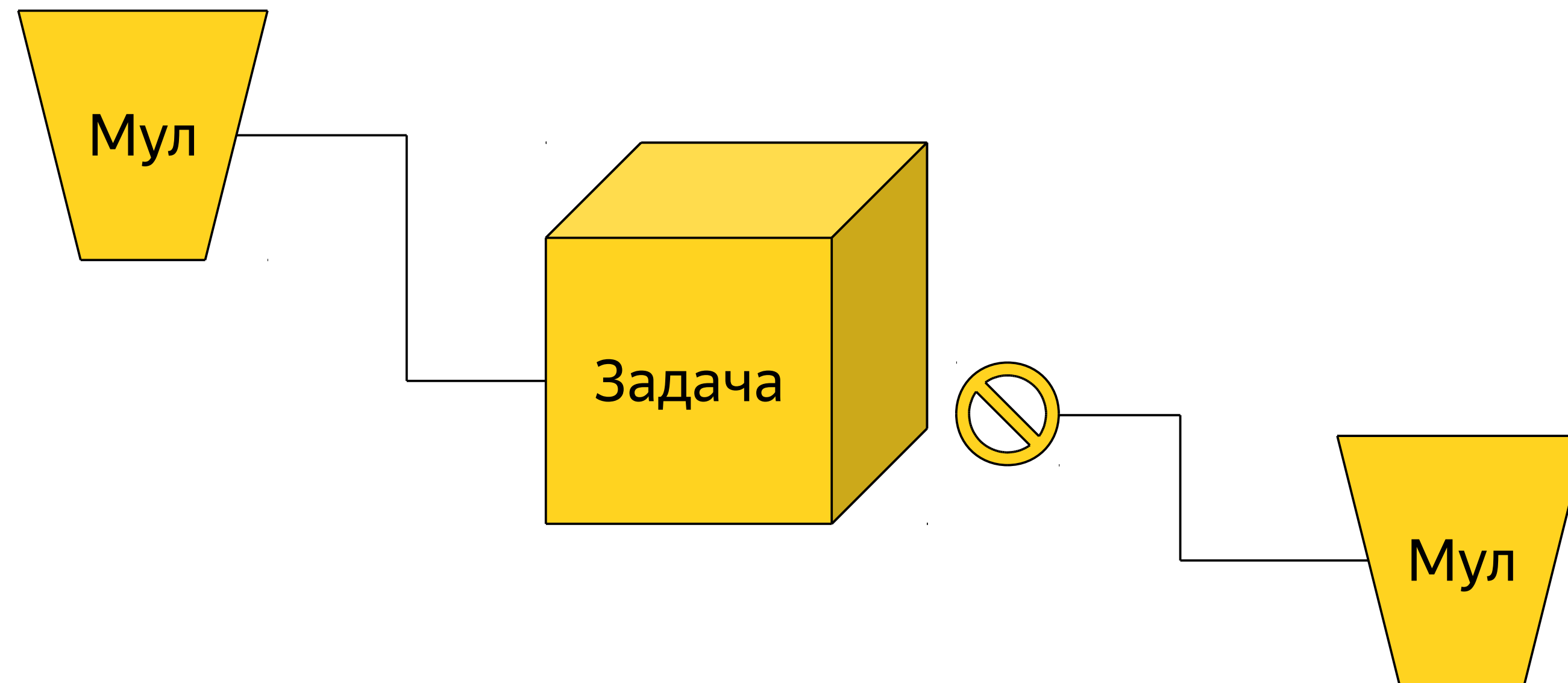


# Кеширование

- На основе хеш-таблиц
- В памяти
- С возможностью сброса в файл
- И доступа по сети
- Поддержка LRU
- Работает с инструментами для маршрутизации

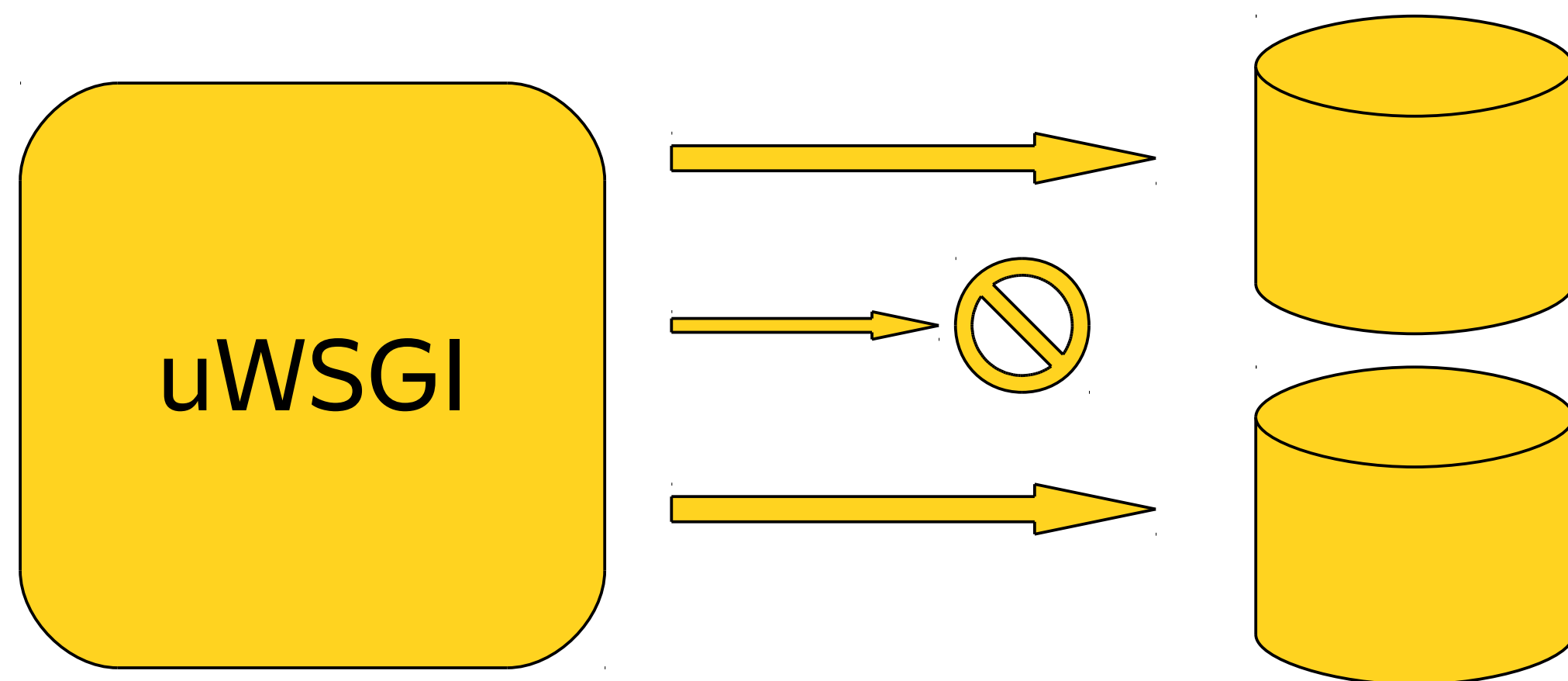
# Блокировки

- Поддержка множества блокировок
- Общий доступ



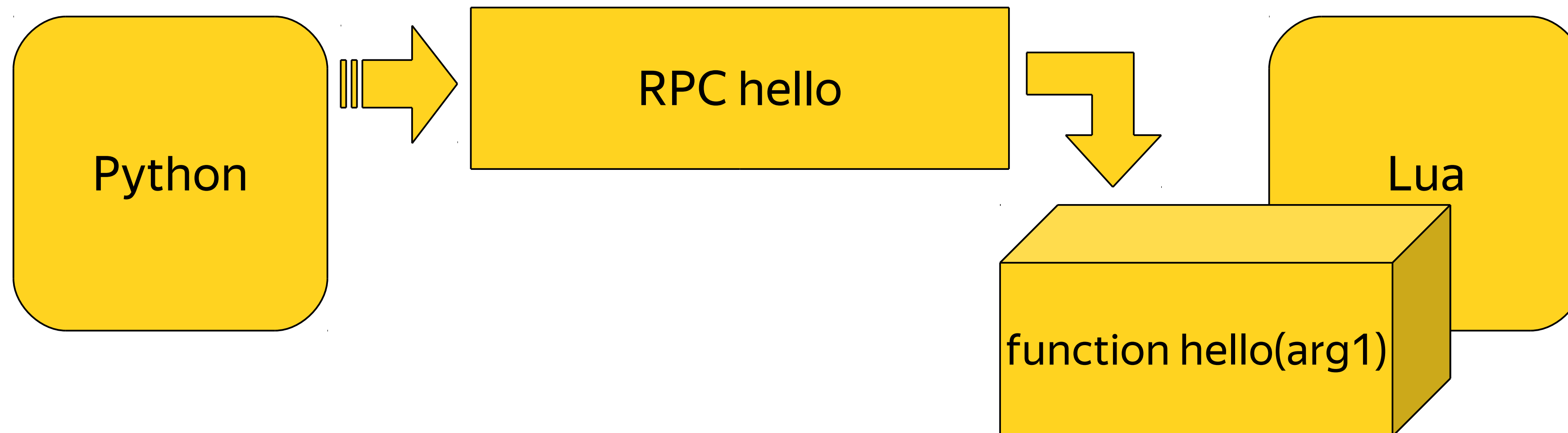
# Протоколирование

- Множество систем назначения: файл, UDP, MongoDB, Redis, rsyslog, ZeroMQ и пр.
- Поддержка форматирования и сжатия
- Ротация по размеру или прикосновению
- Возможность фильтрации записей



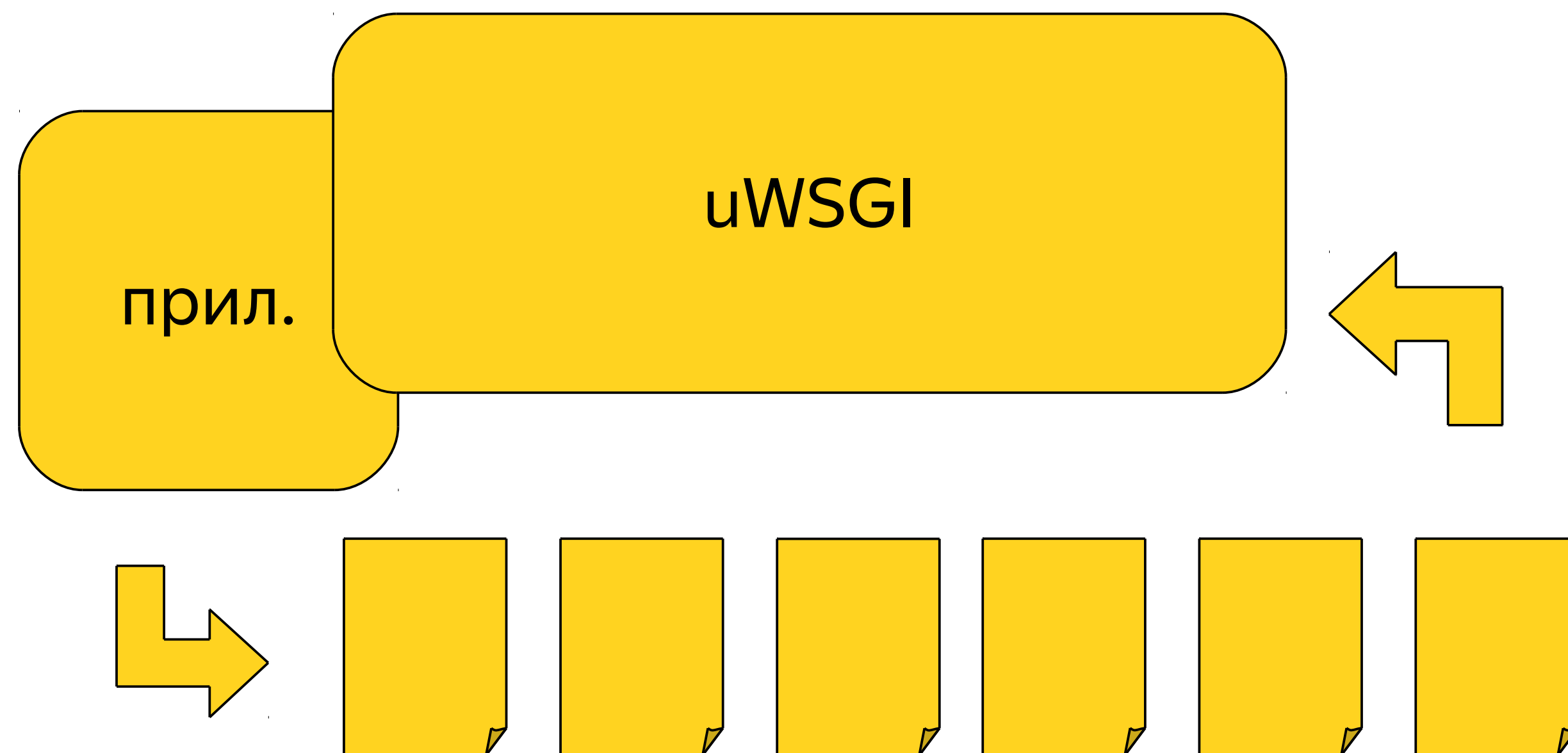
# RPC

- Python, Perl, Java, Lua...
- Простой интерфейс, основанный на передаче и возврате строк



# Планировщик и очередь

- Запуск заданий по таймеру
- Или по расписанию а-ля cron
- Очередь заданий (spooler)

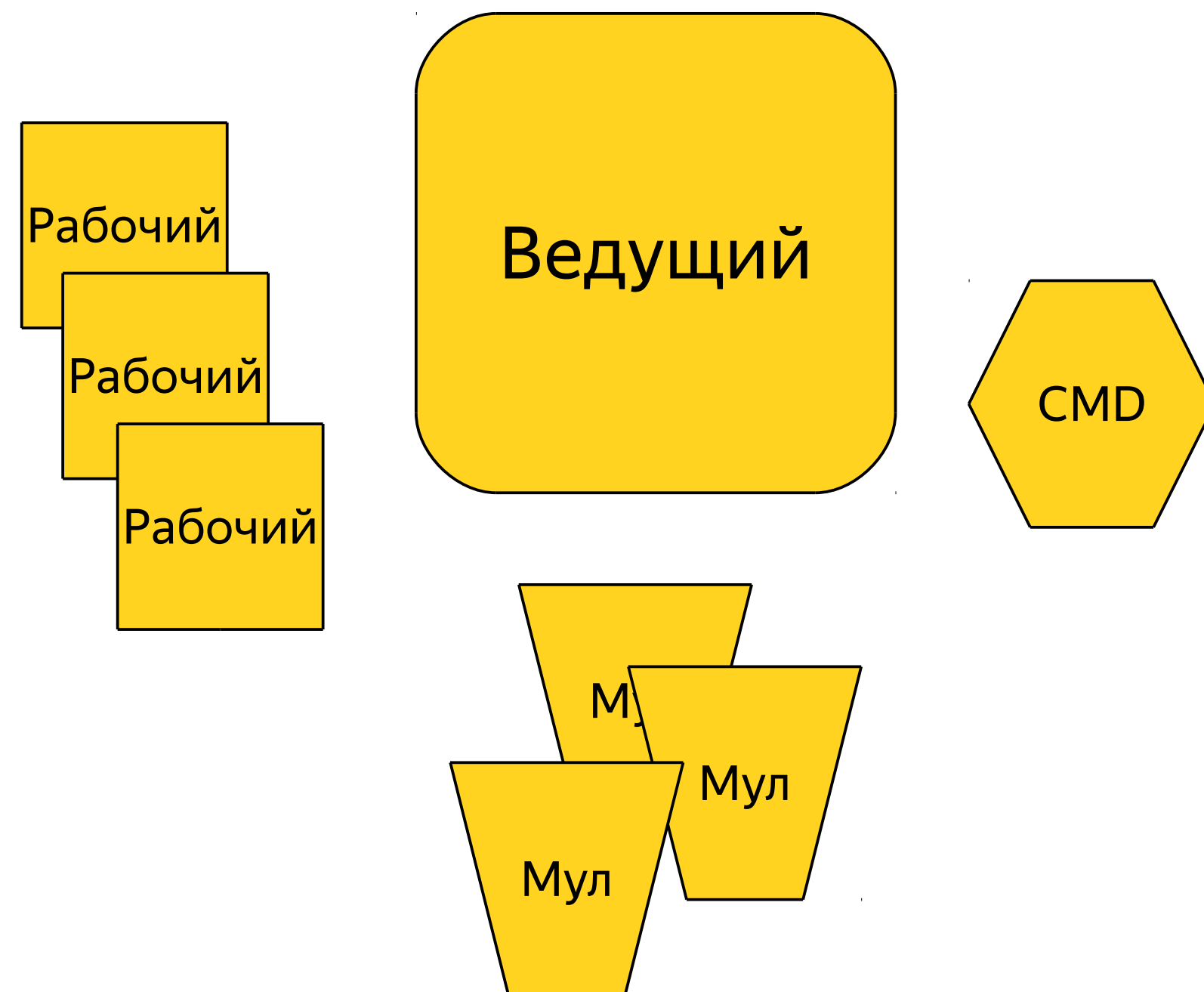


Кроме этого

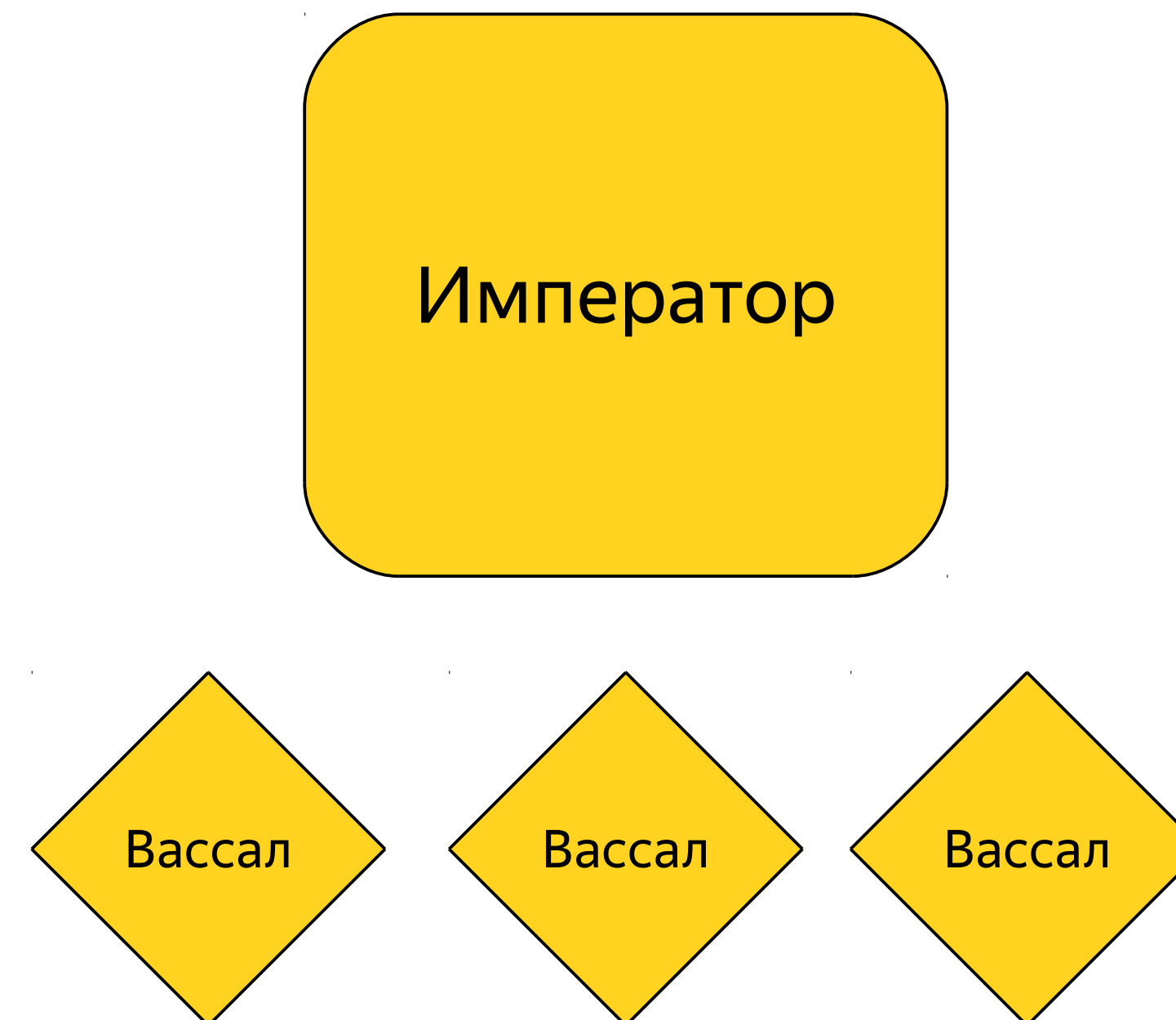


# Управление процессами

- Ведущий процесс

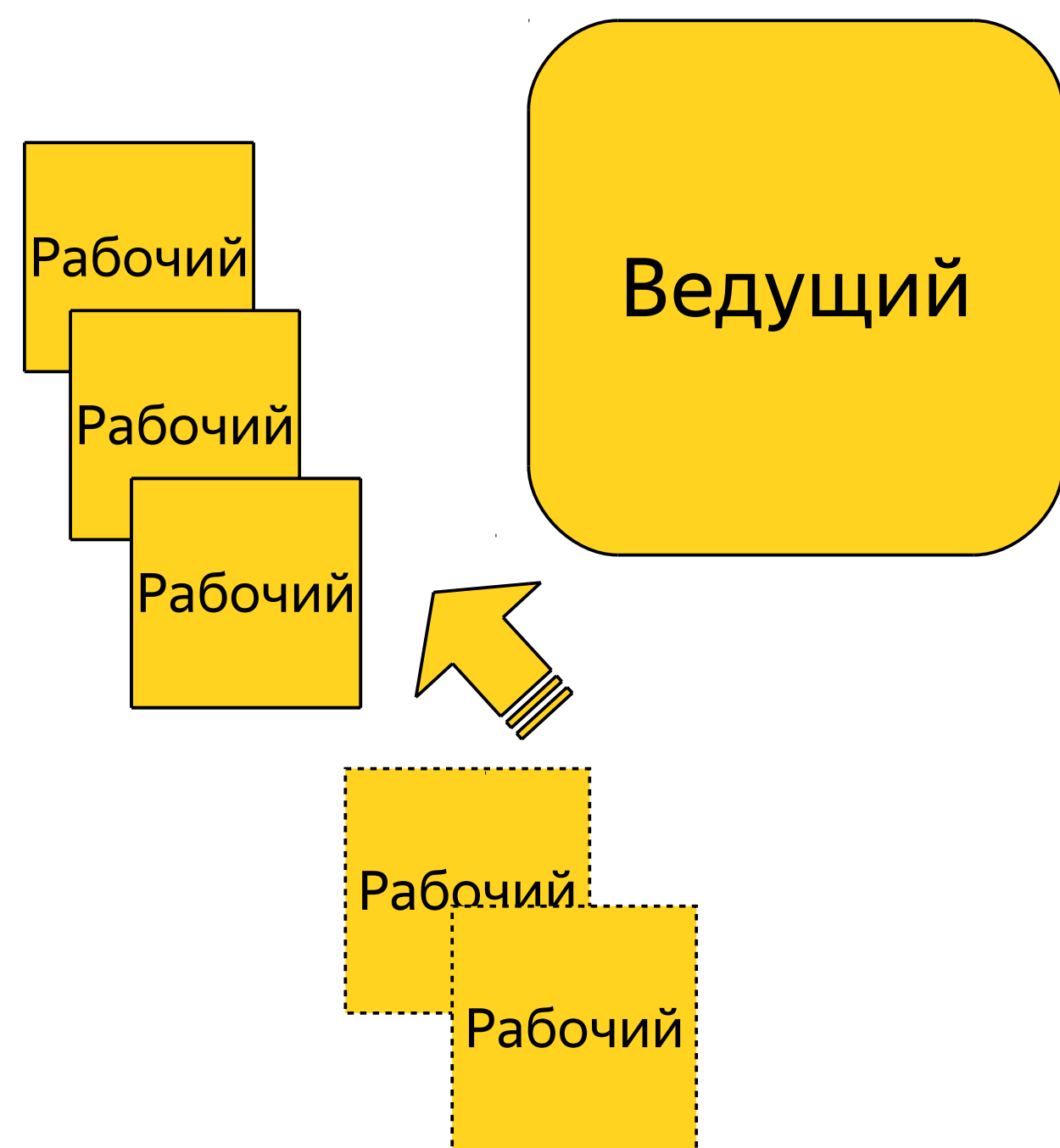


- Император и вассалы

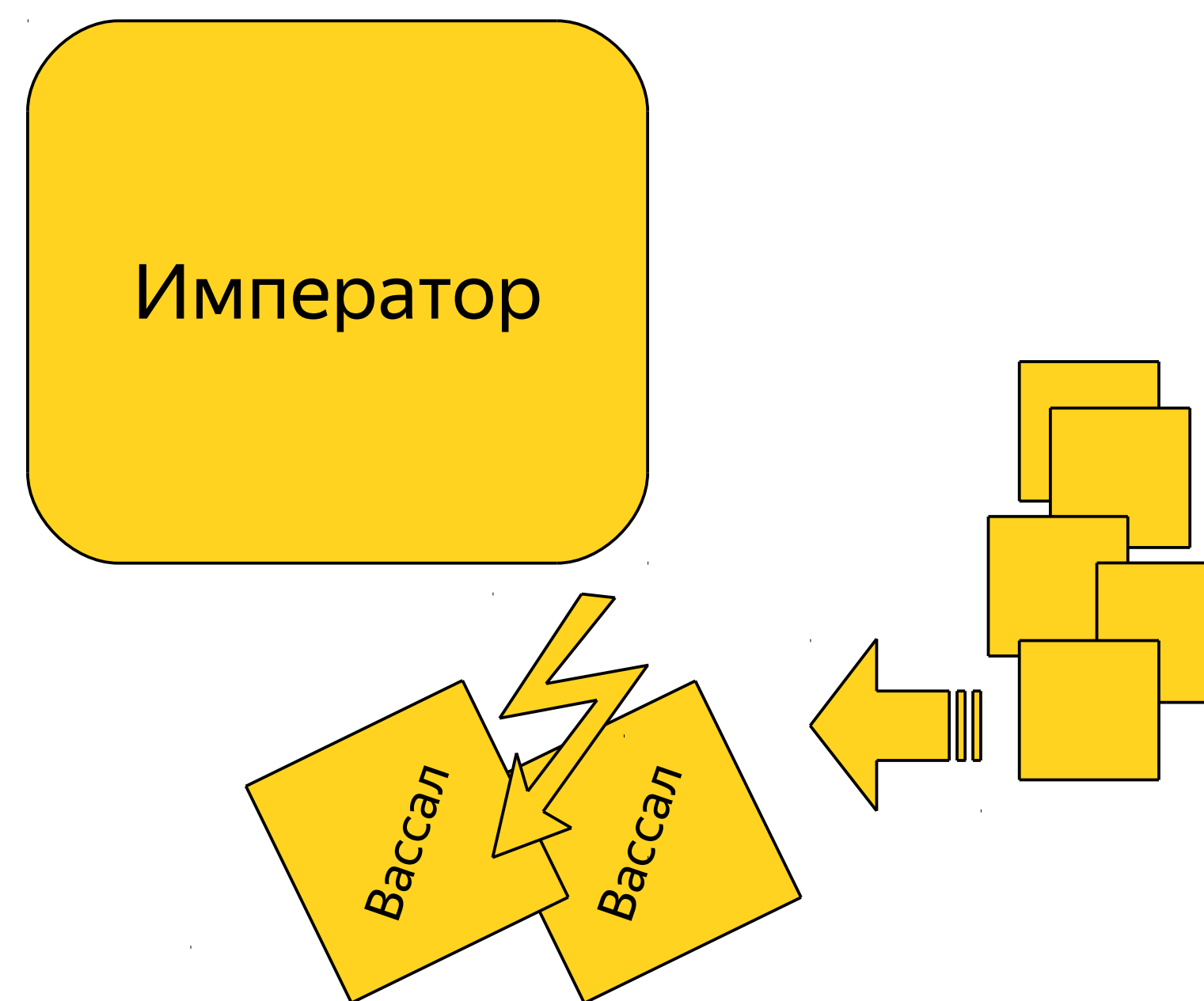


# Адаптация к нагрузке

- Механизм удешевления



- Режим Хозяин стаи — Broodlord для Императора

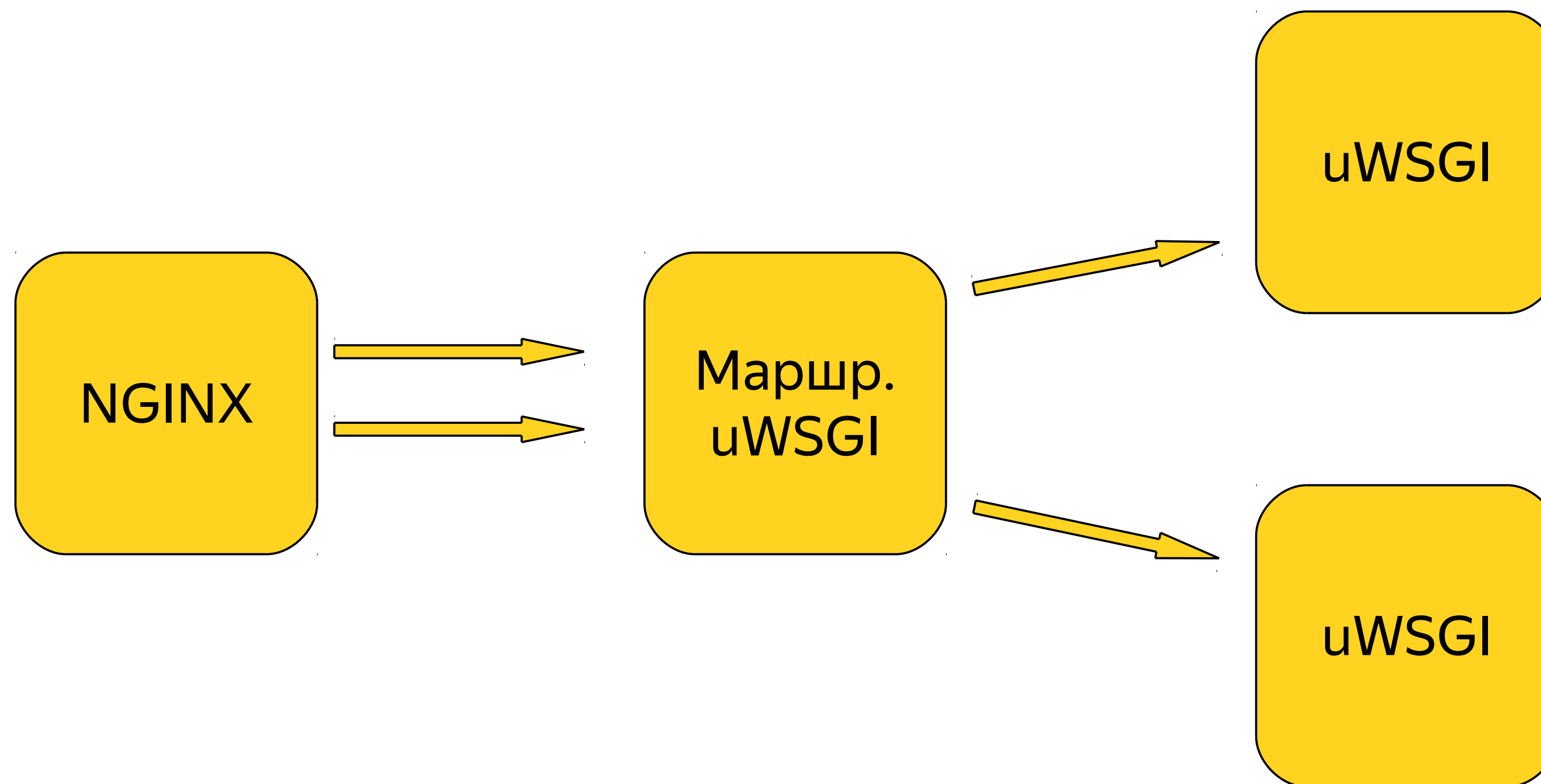




# Маршрутизация, балансировка

- Встроенный маршрутизатор
- Самостоятельные маршрутизаторы:
  - HTTP / HTTPS
  - Fast (протокол uwsgi)
  - Псевдотерминалы
  - TUN / TAP
- Легион — подсистема для кластеризации

# Использование маршрутизатора



800+

параметров настройки





uwsgi**conf**

для упрощения конфигурирования uWSGI

# Конфигурируем с uwsgiconf

```
# uwsgi.py
from uwsgiconf.config import configure_uwsgi
from uwsgiconf.presets.nice import PythonSection

def get_configuration():
    return PythonSection(
        wsgi_module='/home/idle/mywebapp/wsgi.py',

        ).networking.register_socket(
        PythonSection.networking.sockets.http('127.0.0.1:8000'))

configure_uwsgi(get_configuration)

# > uwsgi run
```

# Пример файла на выходе

```
# > uwsgiiconf compile
```

```
[uwsgi]  
plugin = python3  
strict = true  
env = LANG=en_US.UTF-8  
workers = %k  
enable-threads = true  
harakiri-verbose = true  
vacuum = true  
auto-procname = true  
master = true  
die-on-term = true  
thunder-lock = true  
wsgi-file = /home/idle/mywebapp/wsgi.py  
need-app = true  
http-socket = 127.0.0.1:8000
```

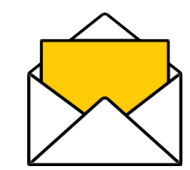
# Обзор

- **Рабочие, мулы и ведущий процесс**
- Кэширование
- Блокировки
- Сигналы и оповещения
- Мониторинг
- Протоколирование
- RPC
- Планировщик и очередь
- Управление процессами
- Адаптация к нагрузке
- Маршрутизация, балансировка
- Веб-сокеты, поддержка асинхронного кода и пр.

# Спасибо!

Игорь Стариков

делаю мир лучше



[idlesign@yandex.ru](mailto:idlesign@yandex.ru)



[pythonz.net](http://pythonz.net)



[idlesign](https://github.com/idlesign)