

Heisenbug

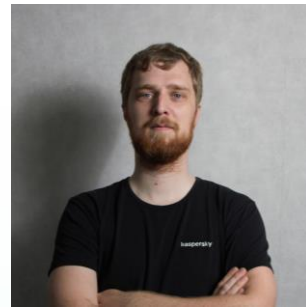
Как сделать удобный инструмент для тестирования операционной системы

Олег Пригода

kaspersky

Кто я такой

Лаборатория Касперского



Олег Пригода

Ведущий разработчик в
тестировании

@prigodaoleg

О чем доклад

Тестирование операционной системы
и продуктов на её основе

Аппаратные стенды

Сборочный конвейер

Инструмент для тестирования

Архитектура инструмента

Примеры тестов

О чем доклад

Тестирование операционной системы
и продуктов на её основе

Аппаратные стенды

Сборочный конвейер

Инструмент для тестирования

Архитектура инструмента

Примеры тестов

О чем доклад

Тестирование операционной системы
и продуктов на её основе

Аппаратные стенды

Сборочный конвейер

Инструмент для тестирования

Архитектура инструмента

Примеры тестов

О чем доклад

Тестирование операционной системы
и продуктов на её основе

Аппаратные стенды

Сборочный конвейер

Инструмент для тестирования

Архитектура инструмента

Примеры тестов

О чем доклад

Тестирование операционной системы
и продуктов на её основе

Аппаратные стенды

Сборочный конвейер

Инструмент для тестирования

Архитектура инструмента

Примеры тестов

О чем доклад

Тестирование операционной системы
и продуктов на её основе

Аппаратные стенды

Сборочный конвейер

Инструмент для тестирования

Архитектура инструмента

Примеры тестов

KasperskyOS – это не Linux

Собственная разработка
Микроядро + компоненты

Тестов на эмуляторах недостаточно
Необходимо тестировать работу на реальном железе

Важно автоматизировать как можно больше

Иначе не удастся регулярно выпускать релизы для такого огромного проекта

Автотесты пишут не программисты
Среди тестировщиков есть радио-инженеры

KasperskyOS – это не Linux

Собственная разработка
Микроядро + компоненты

Тестов на эмуляторах недостаточно
Необходимо тестировать работу на
реальном железе

**Важно автоматизировать как можно
больше**
Иначе не удастся регулярно выпускать
релизы для такого огромного проекта

Автотесты пишут не программисты
Среди тестировщиков есть радио-
инженеры

KasperskyOS – это не Linux

Собственная разработка
Микроядро + компоненты

Тестов на эмуляторах недостаточно
Необходимо тестировать работу на
реальном железе

**Важно автоматизировать как можно
больше**
Иначе не удастся регулярно выпускать
релизы для такого огромного проекта

Автотесты пишут не программисты
Среди тестировщиков есть радио-
инженеры

KasperskyOS – это не Linux

Собственная разработка
Микроядро + компоненты

Тестов на эмуляторах недостаточно
Необходимо тестировать работу на
реальном железе

**Важно автоматизировать как можно
больше**

Иначе не удастся регулярно выпускать
релизы для такого огромного проекта

Автотесты пишут не программисты
Среди тестировщиков есть радио-
инженеры, инженеры-электроники,
системные администраторы

Какими могут быть стенды

Реальная машина

На ней выполняется
внутренний код теста

Виртуальная машина

Точка входа в тест, управляет
тестовым сценарием

Инструменты

Взаимодействие с реальной
машиной

Другие виртуальные машины

Сервера, целевые машины для
подключения по RDP

Какими могут быть стенды

Реальная машина

На ней выполняется
внутренний код теста

Виртуальная машина

Точка входа в тест, управляет
тестовым сценарием

Инструменты

Взаимодействие с реальной
машиной

Другие виртуальные машины

Сервера, целевые машины для
подключения по RDP

Какими могут быть стенды

Реальная машина

На ней выполняется
внутренний код теста

Виртуальная машина

Точка входа в тест, управляет
тестовым сценарием

Инструменты

Взаимодействие с реальной
машиной

Другие виртуальные машины

Сервера, целевые машины для
подключения по RDP

Какими могут быть стенды

Реальная машина

На ней выполняется
внутренний код теста

Виртуальная машина

Точка входа в тест, управляет
тестовым сценарием

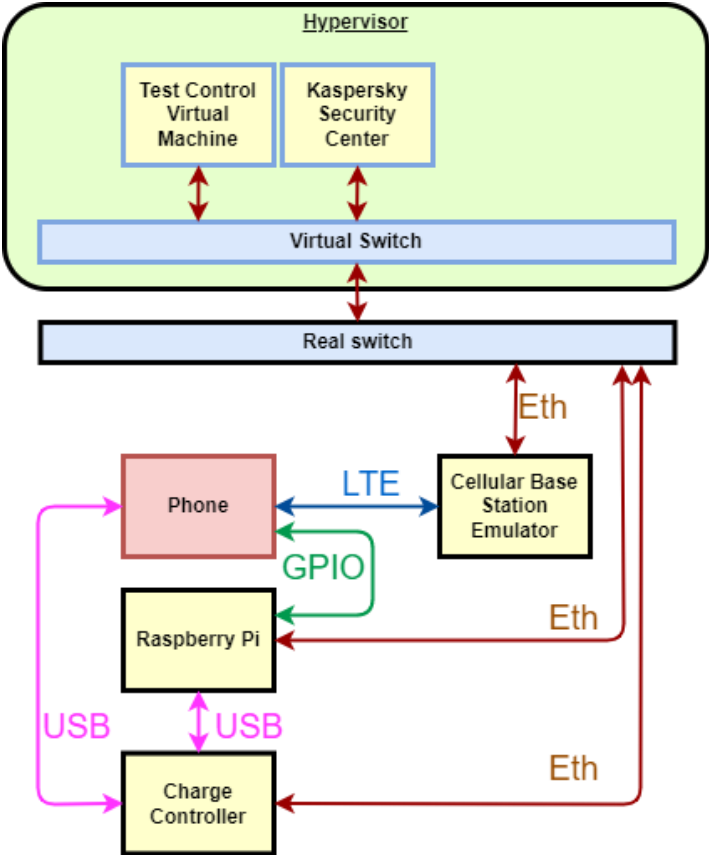
Инструменты

Взаимодействие с реальной
машиной

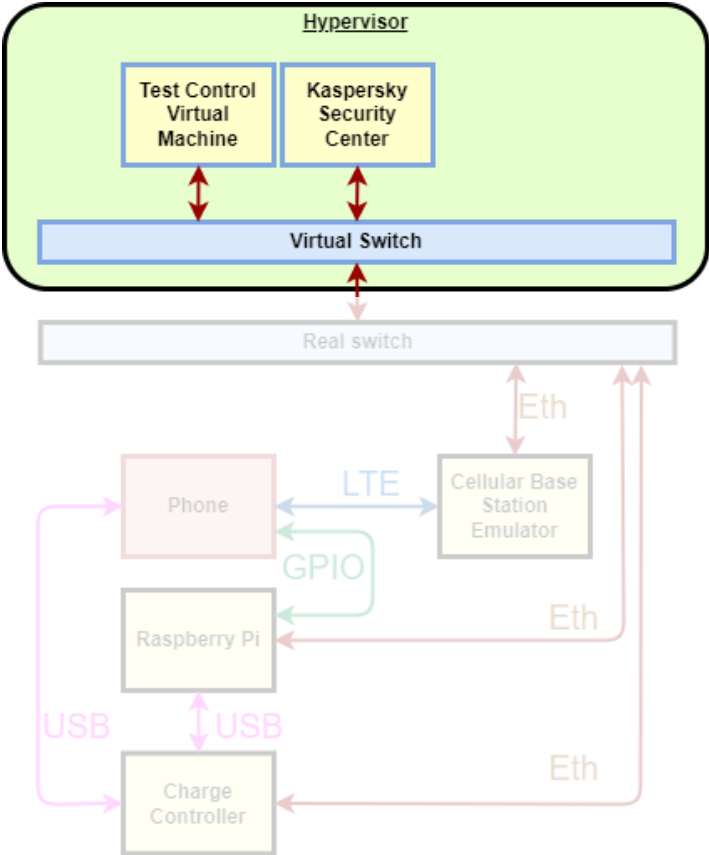
Другие виртуальные машины

Сервера, целевые машины для
подключения по RDP

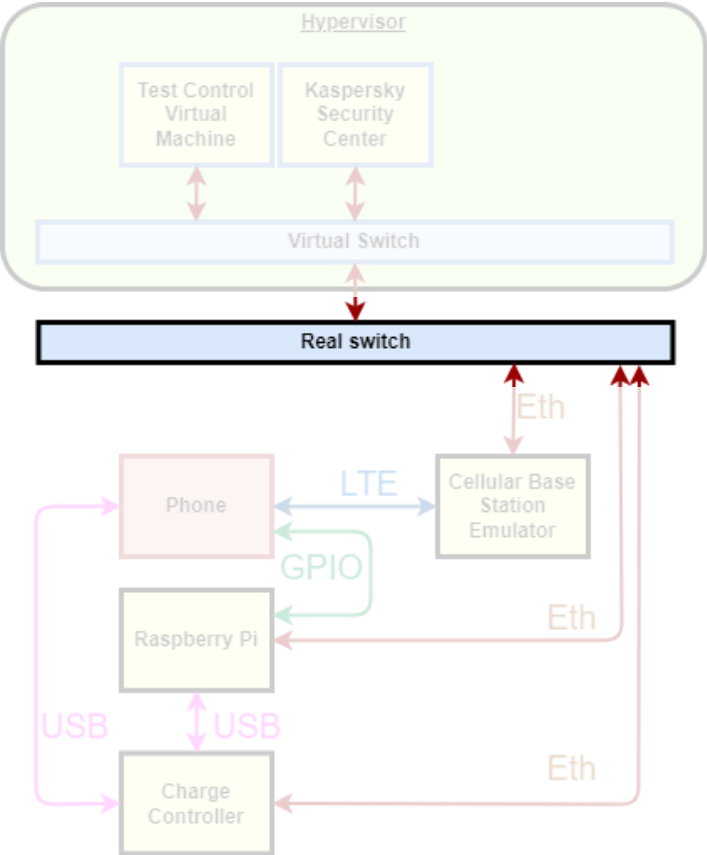
Стенд для тестирования мобильной ОС



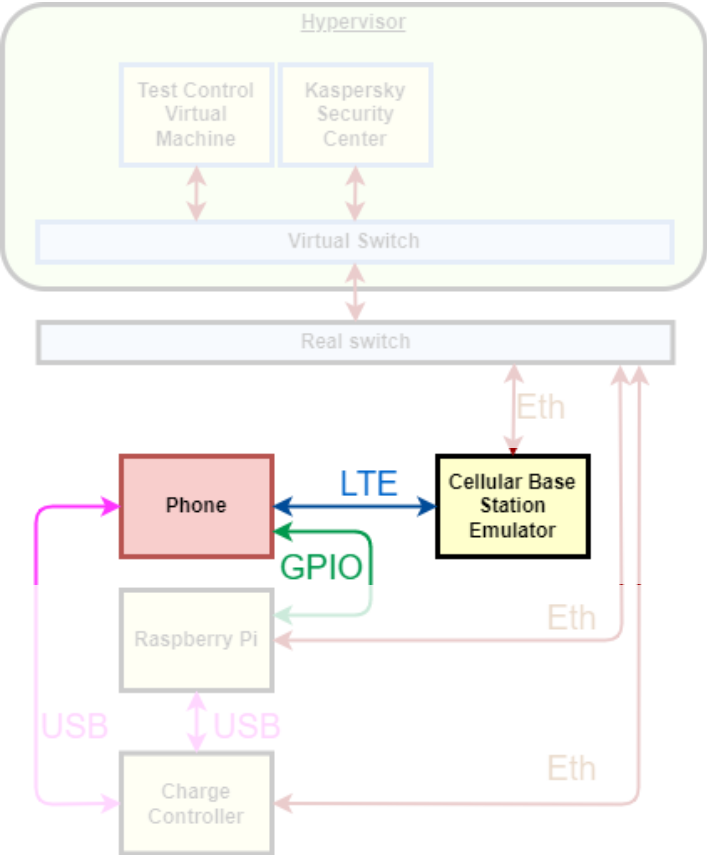
Стенд для тестирования мобильной ОС



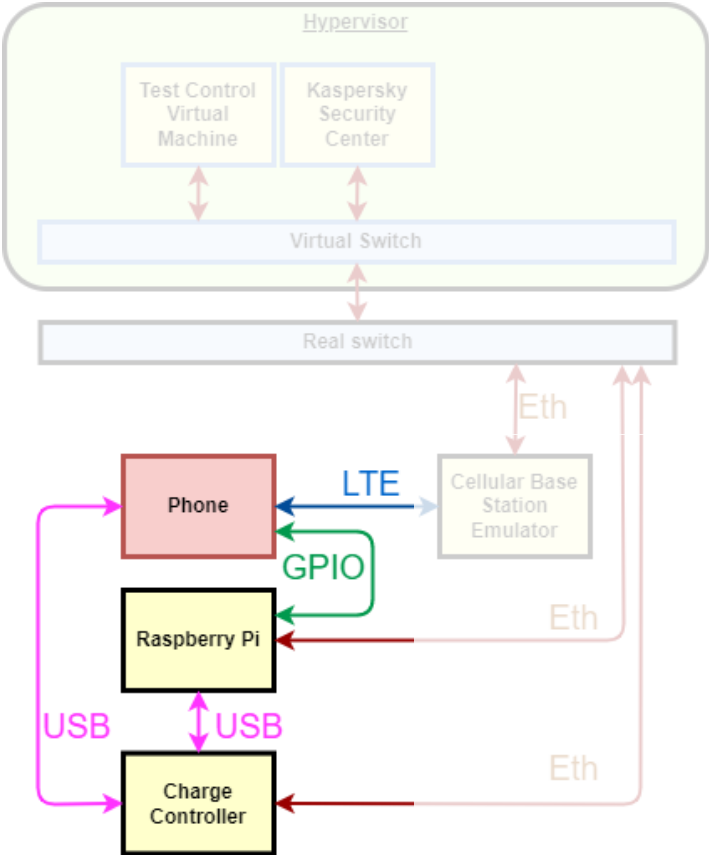
Стенд для тестирования мобильной ОС



Стенд для тестирования мобильной ОС

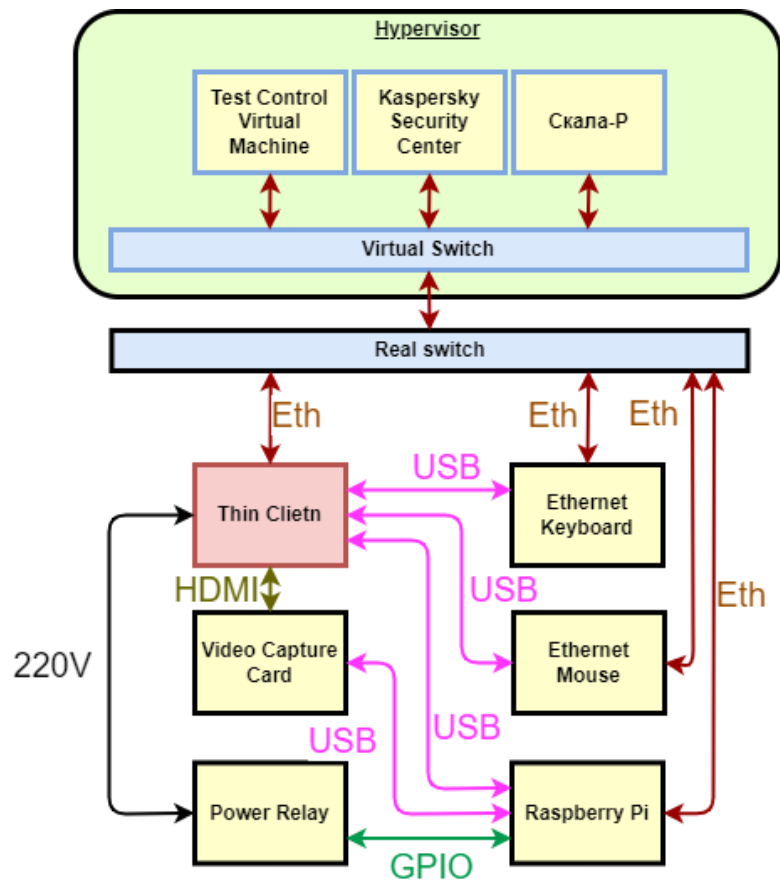


Стенд для тестирования мобильной ОС

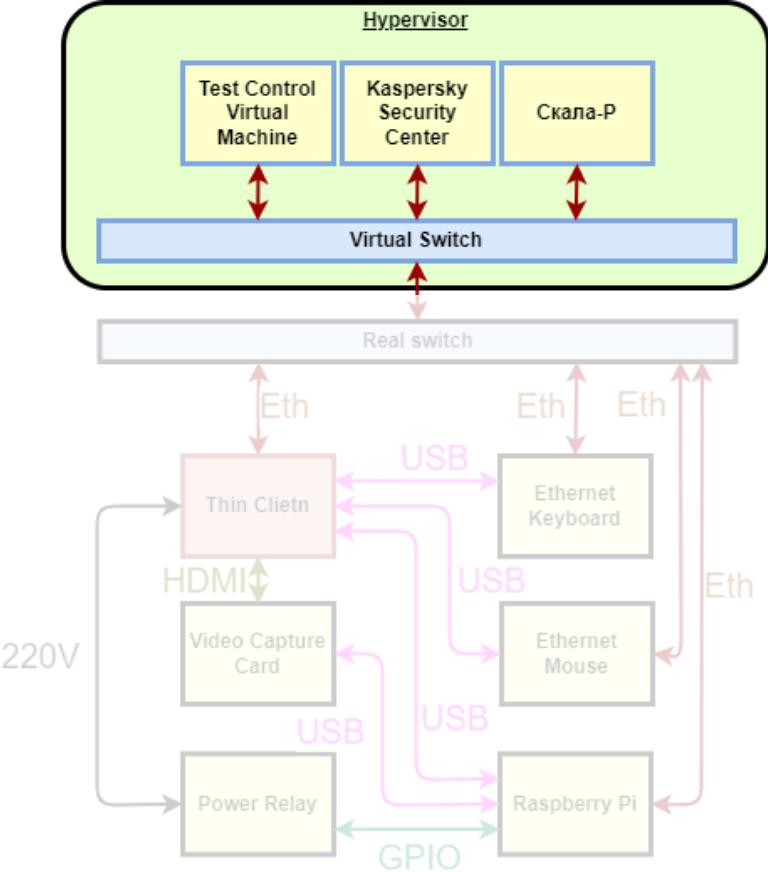


Стенд для тестирования тонкого клиента

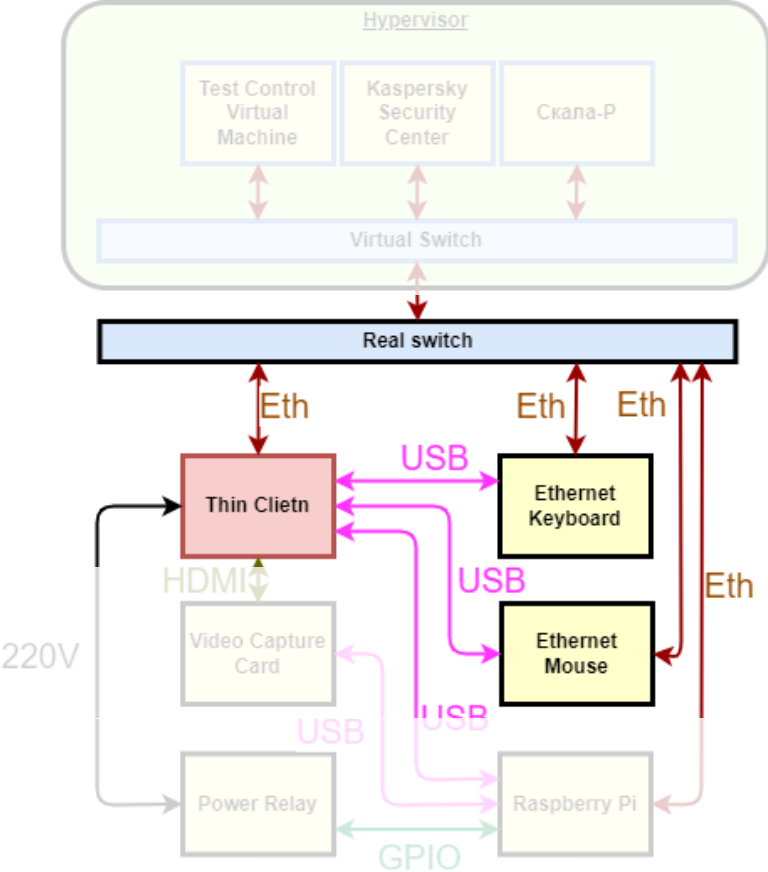
22



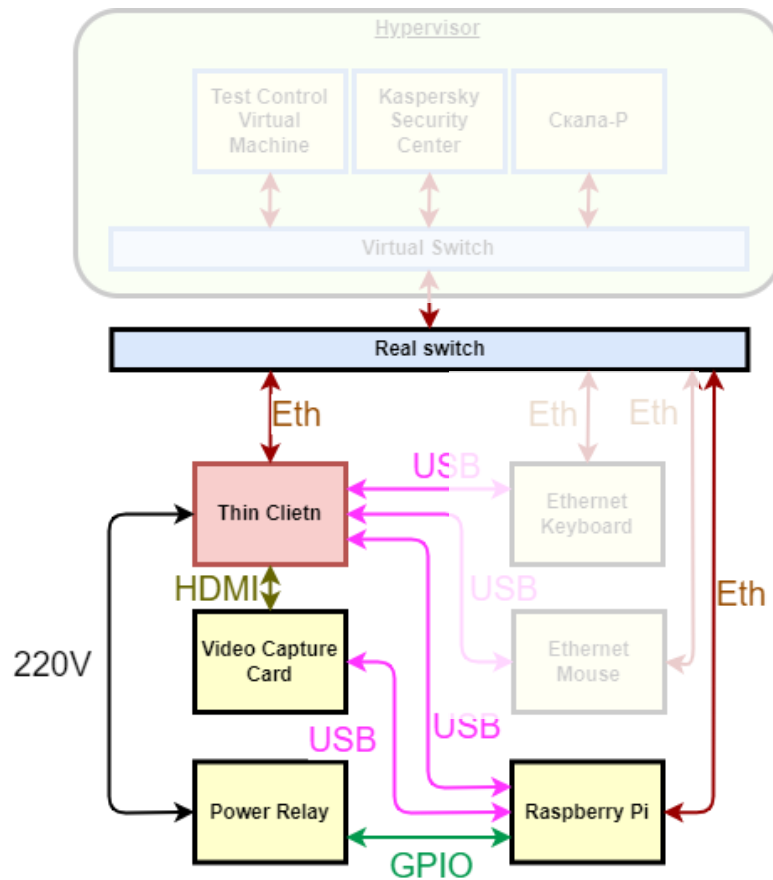
Стенд для тестирования тонкого клиента



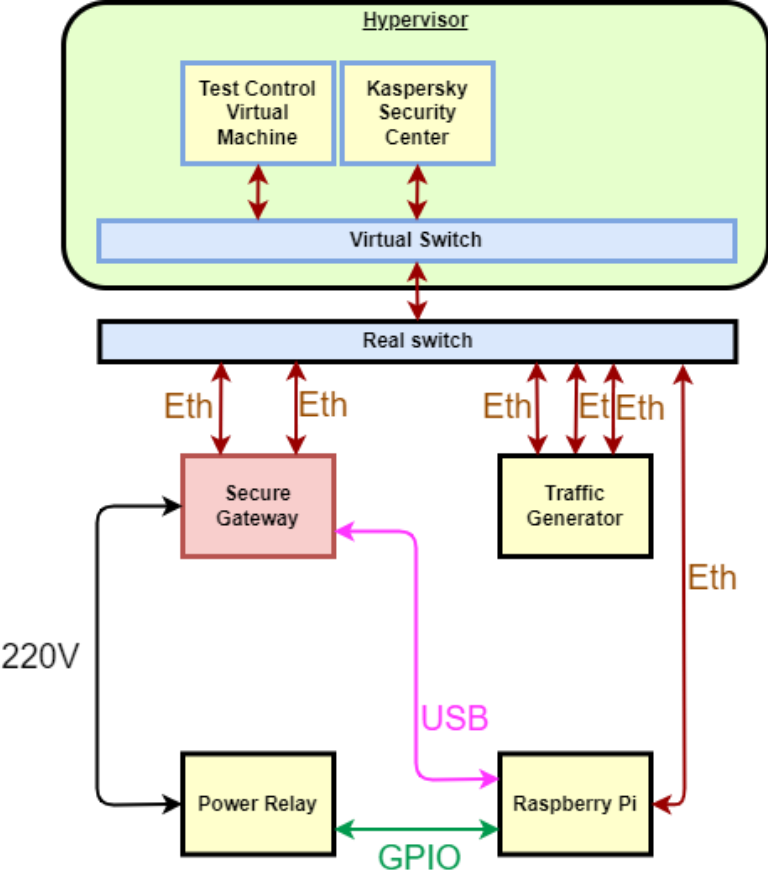
Стенд для тестирования тонкого клиента



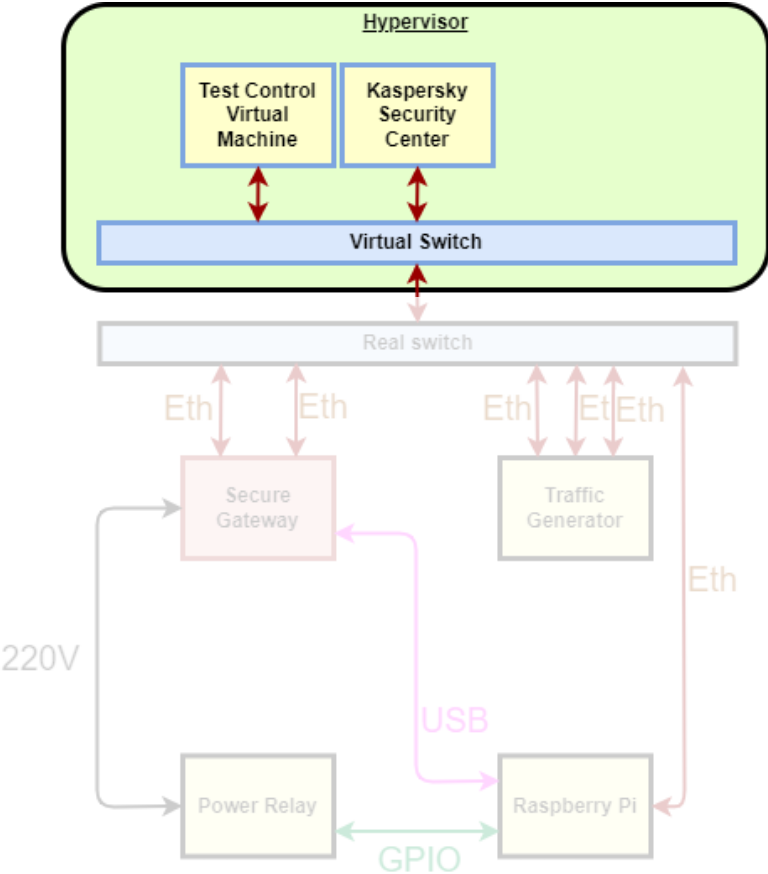
Стенд для тестирования тонкого клиента



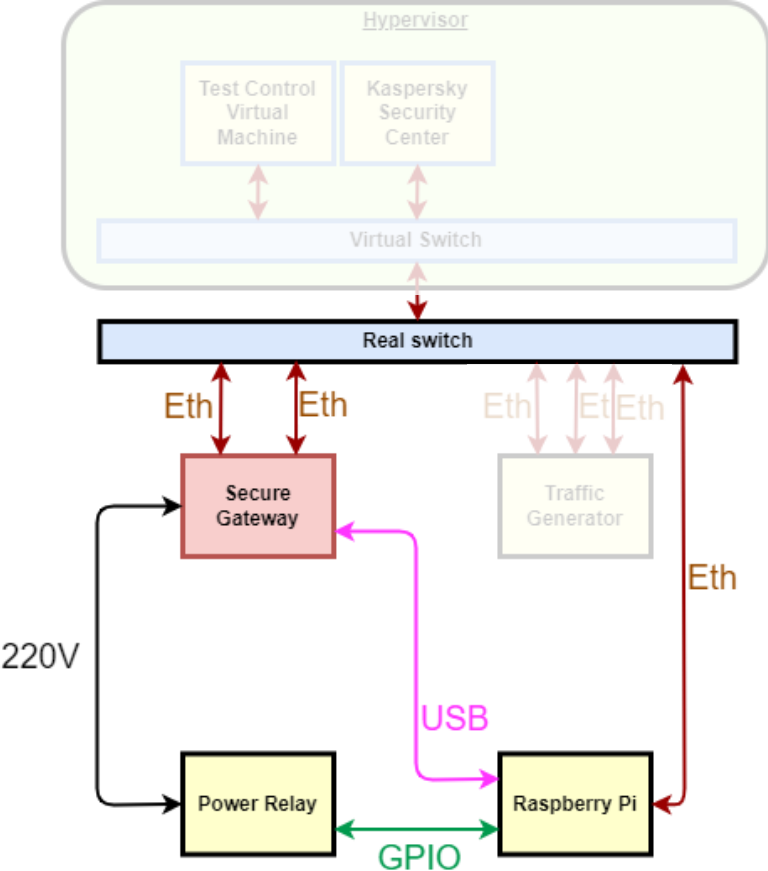
Стенд для тестирования безопасного шлюза



Стенд для тестирования безопасного шлюза

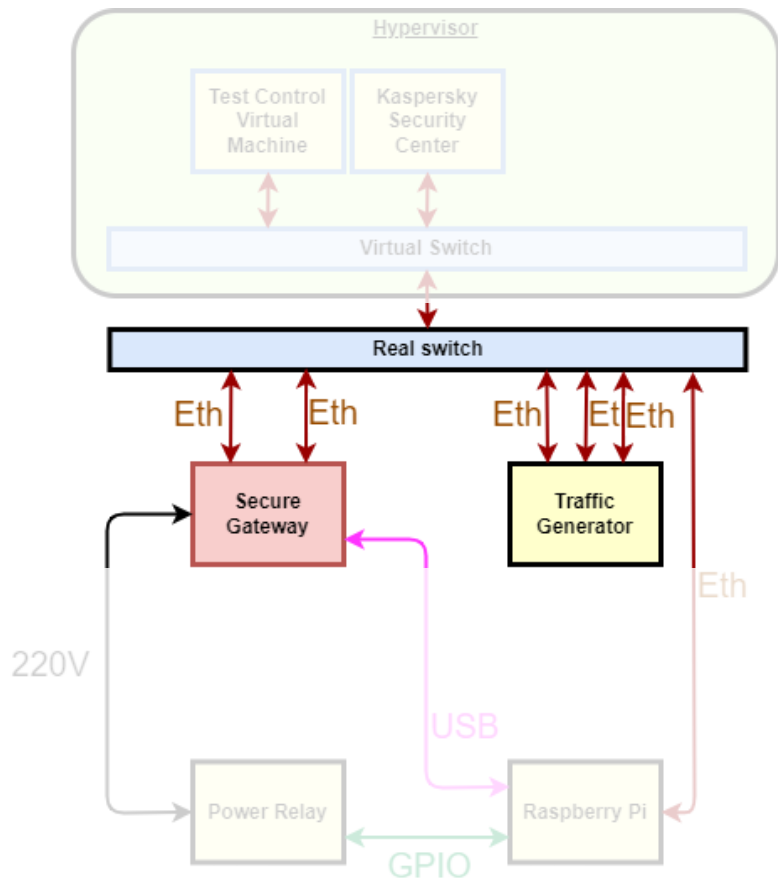


Стенд для тестирования безопасного шлюза

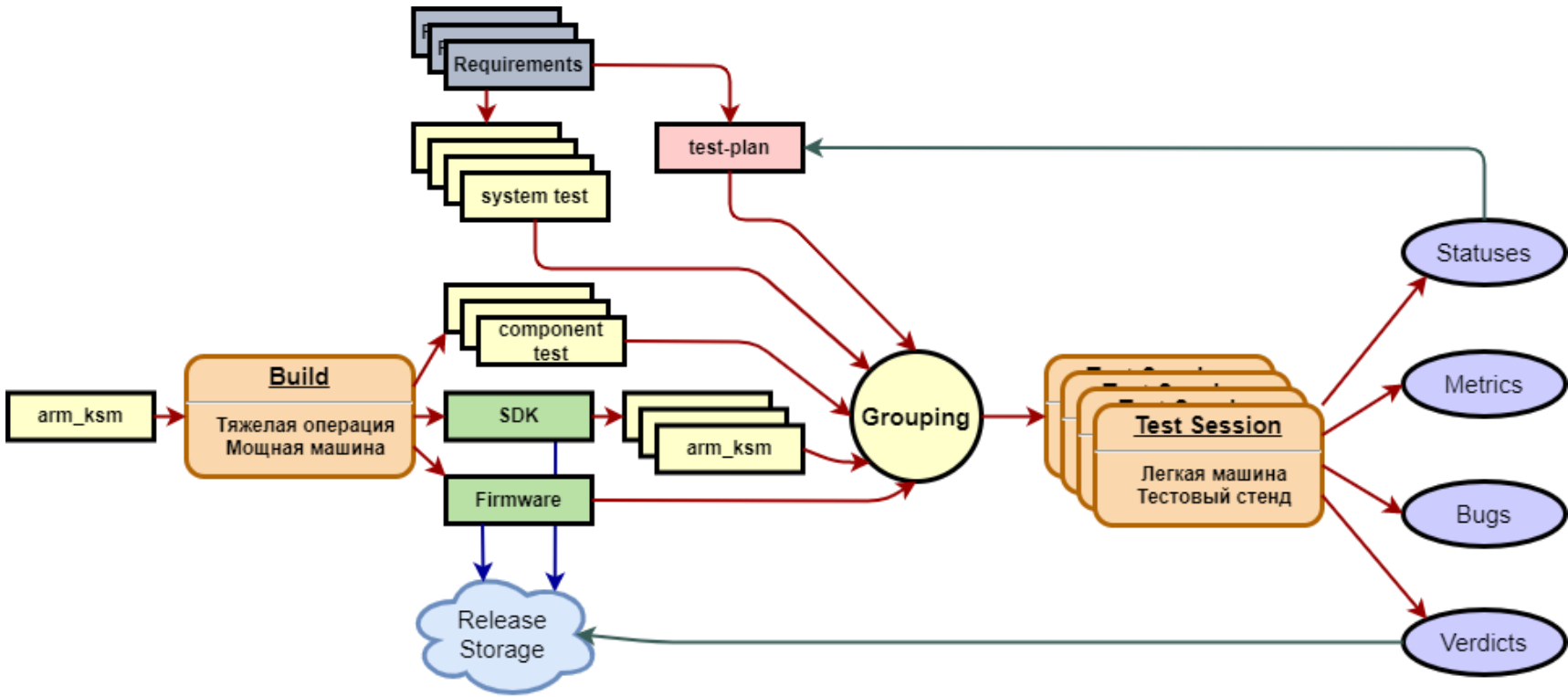


Стенд для тестирования безопасного шлюза

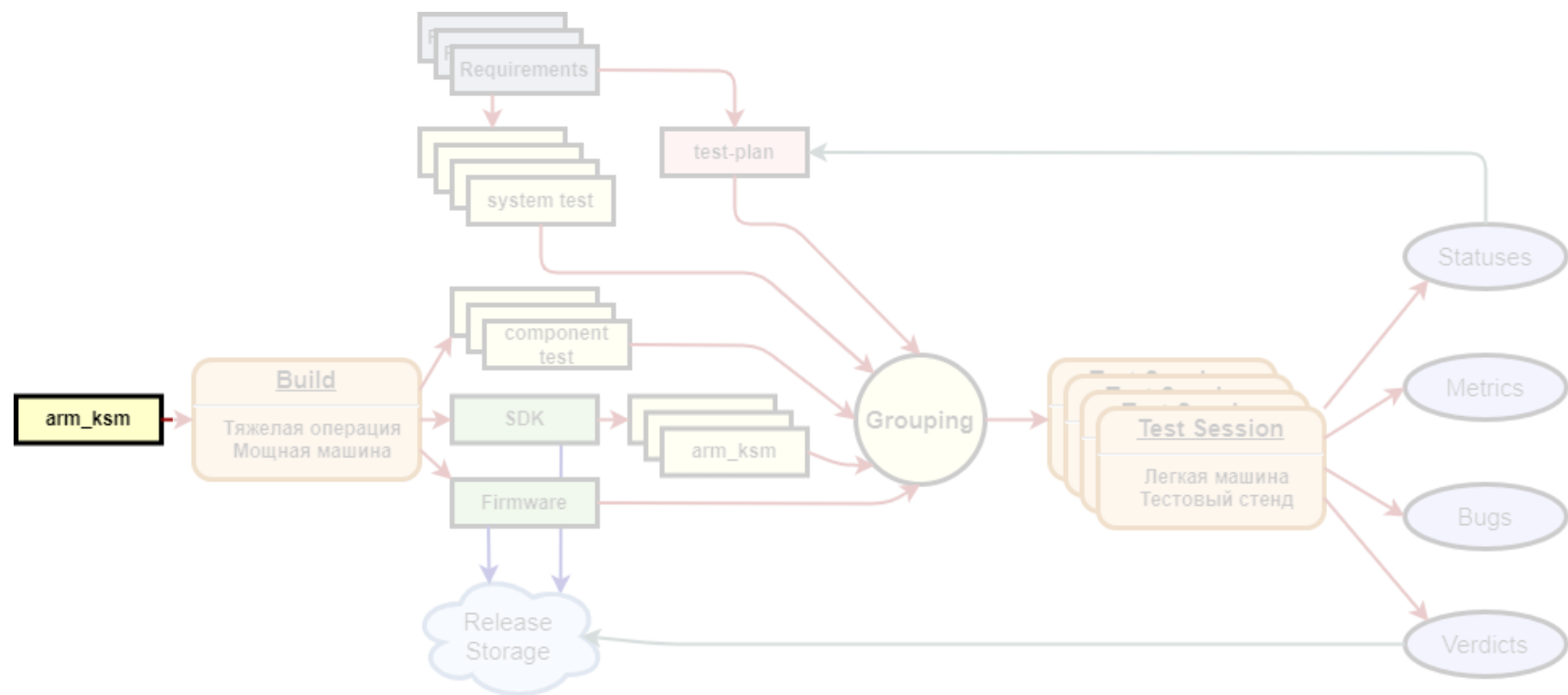
29



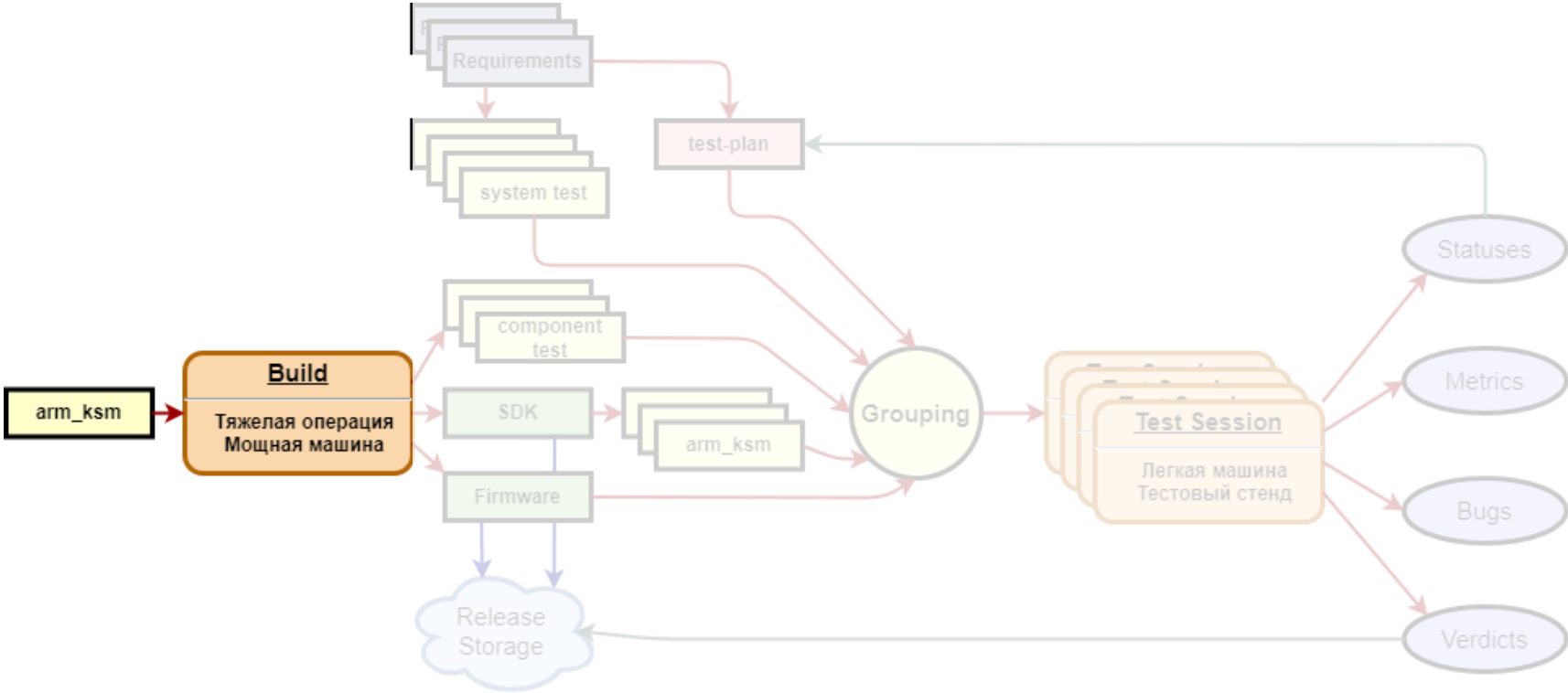
Примерная схема сборочного конвейера



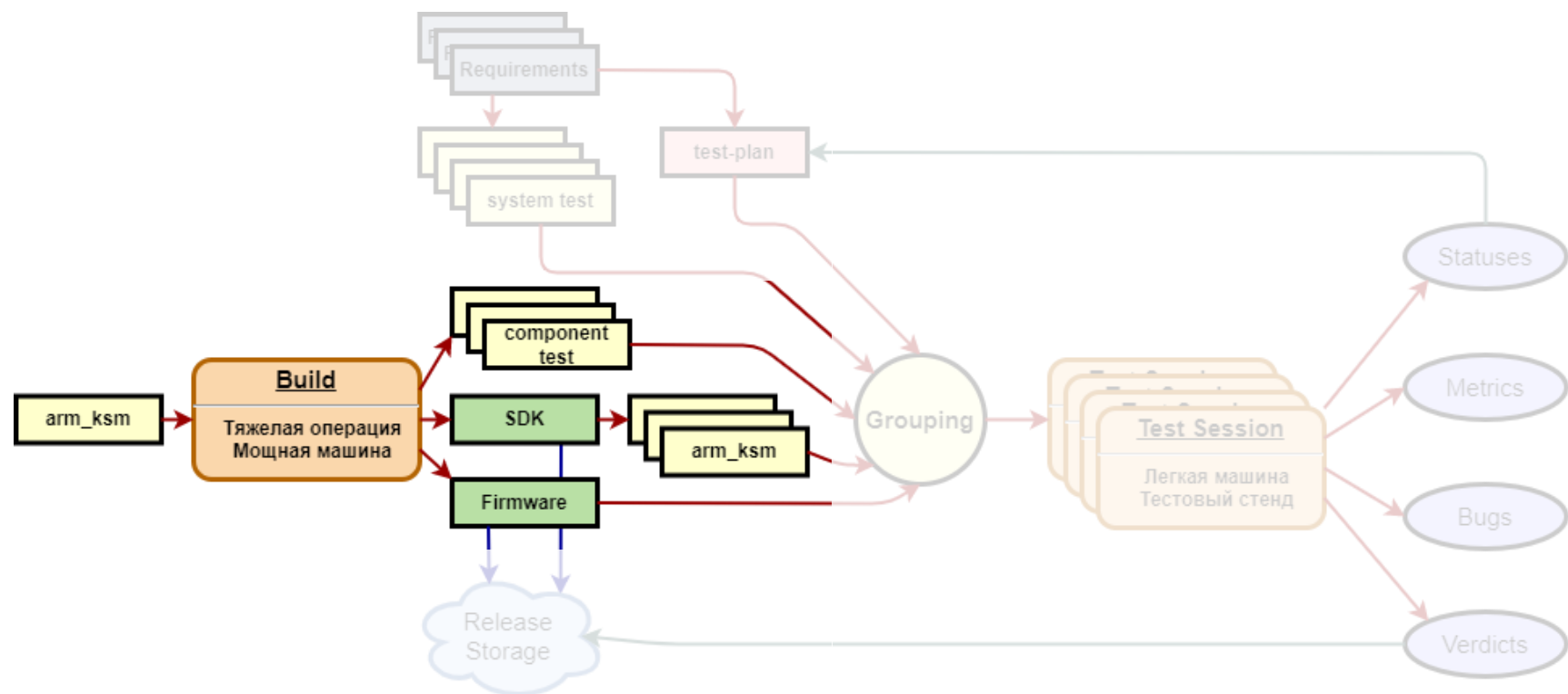
Примерная схема сборочного конвейера



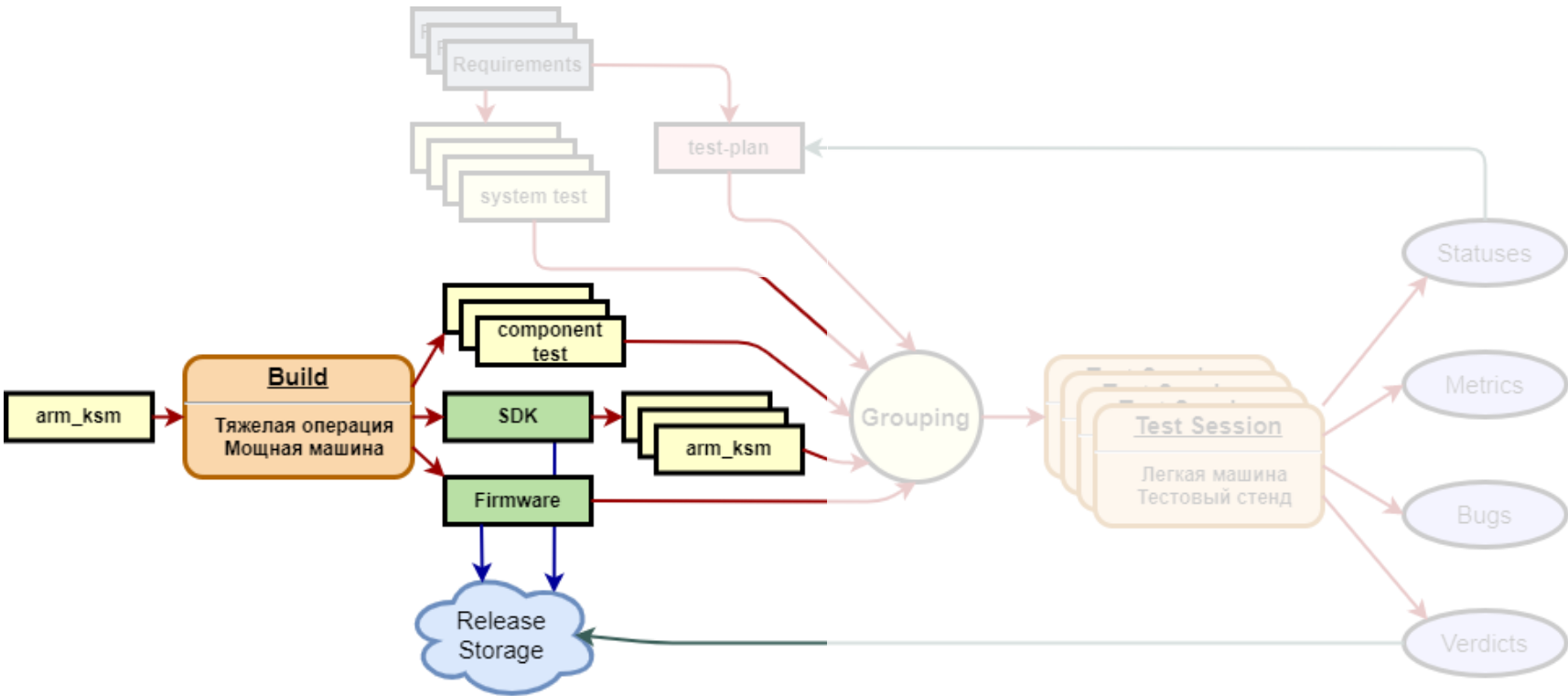
Примерная схема сборочного конвейера



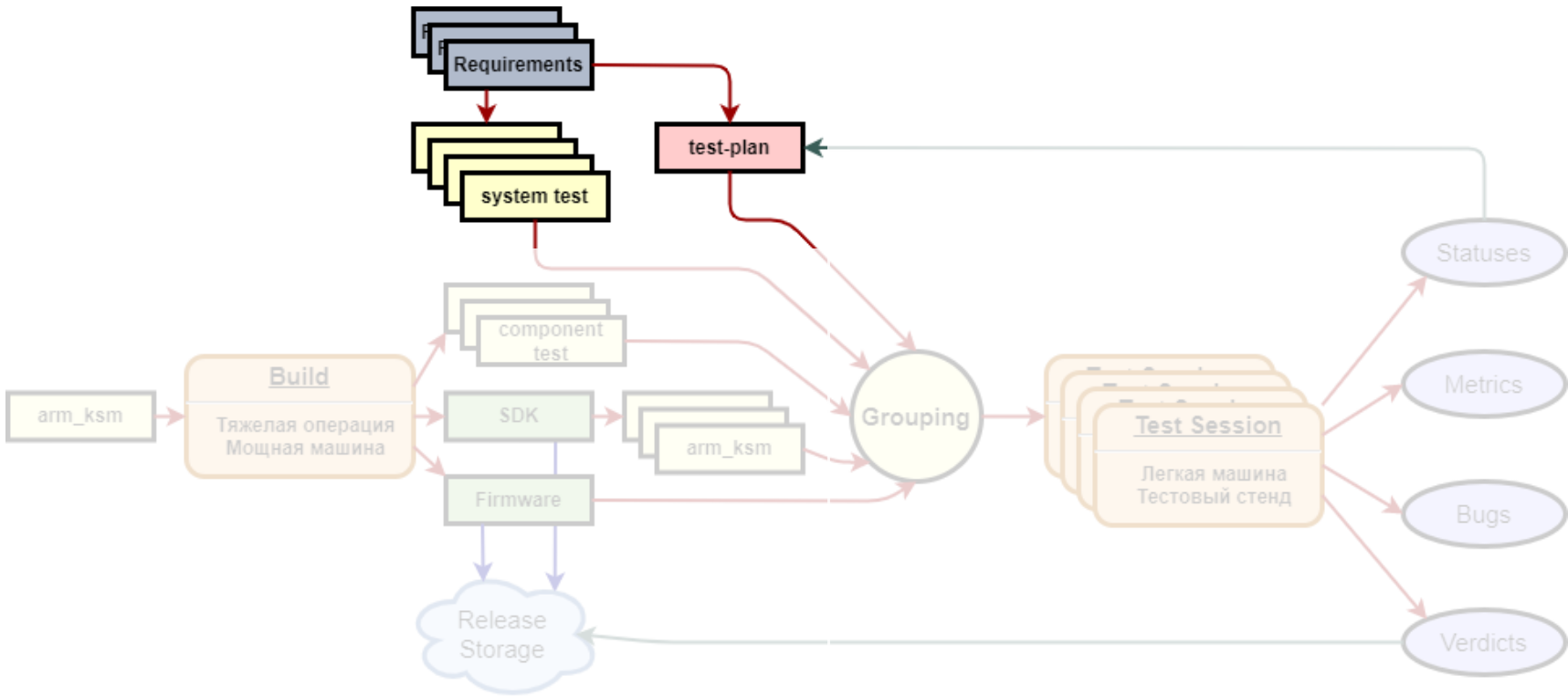
Примерная схема сборочного конвейера



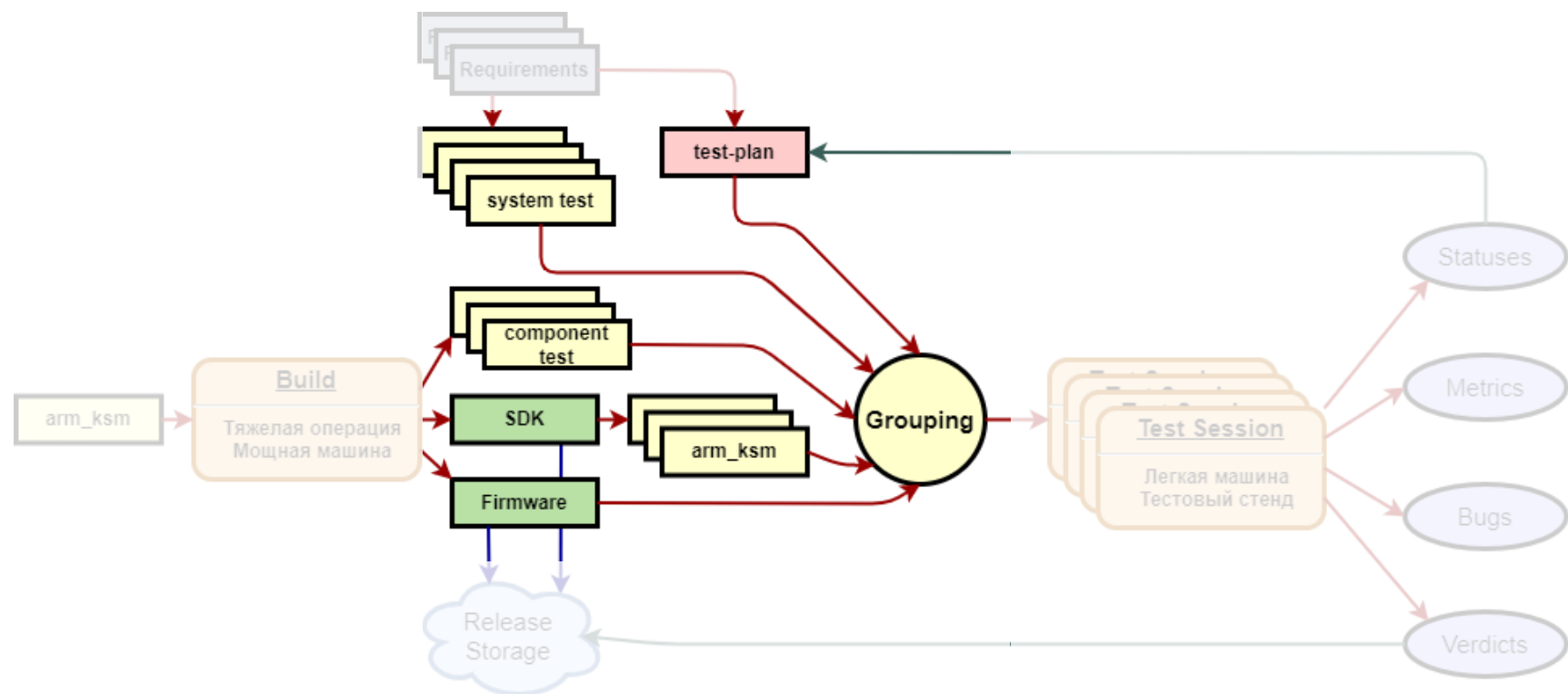
Примерная схема сборочного конвейера



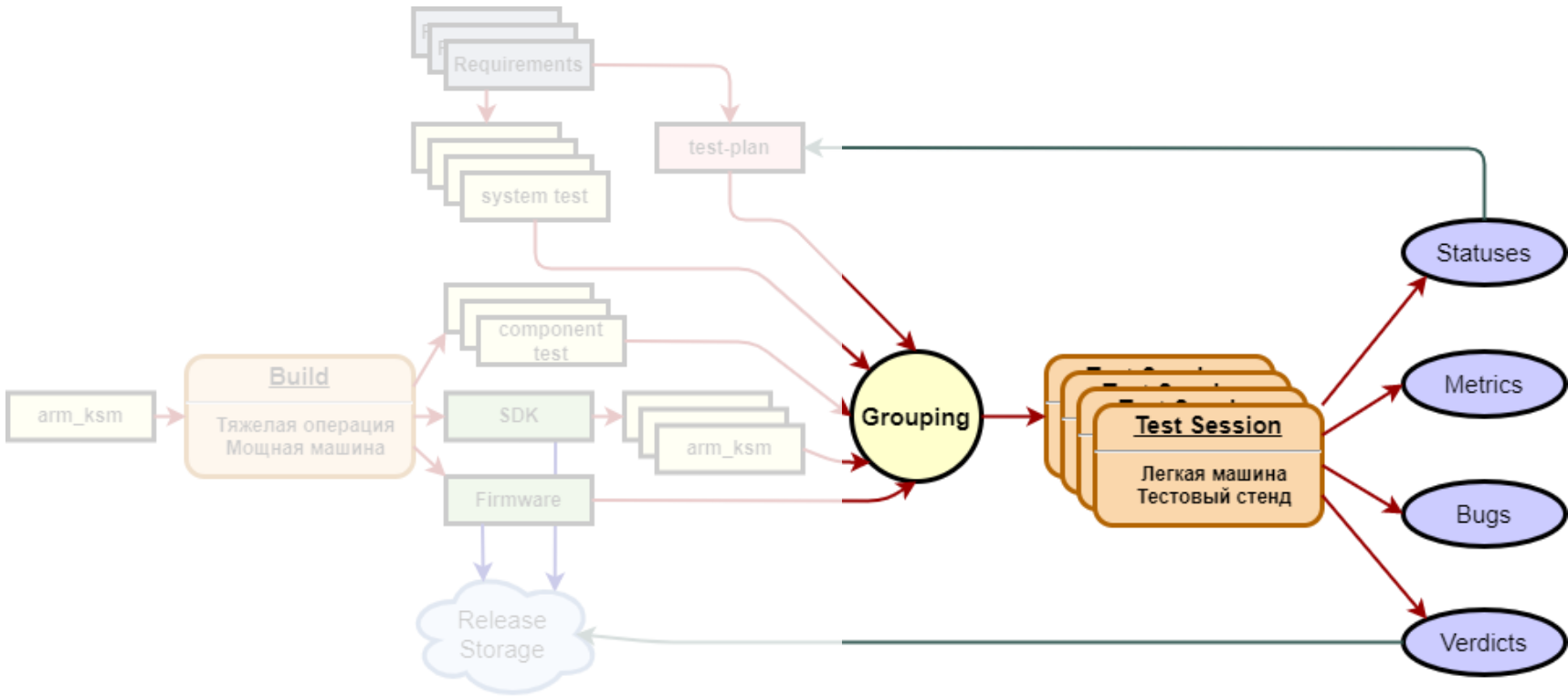
Примерная схема сборочного конвейера



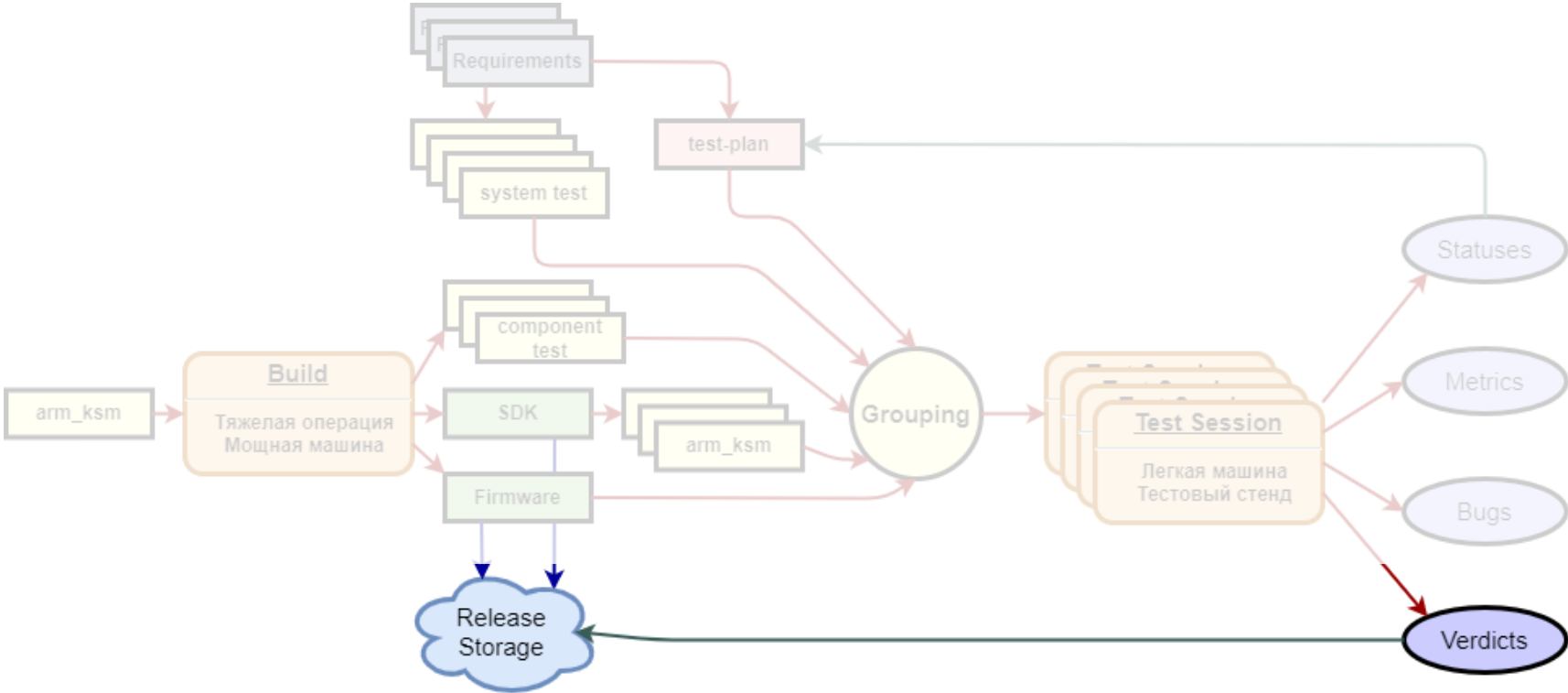
Примерная схема сборочного конвейера



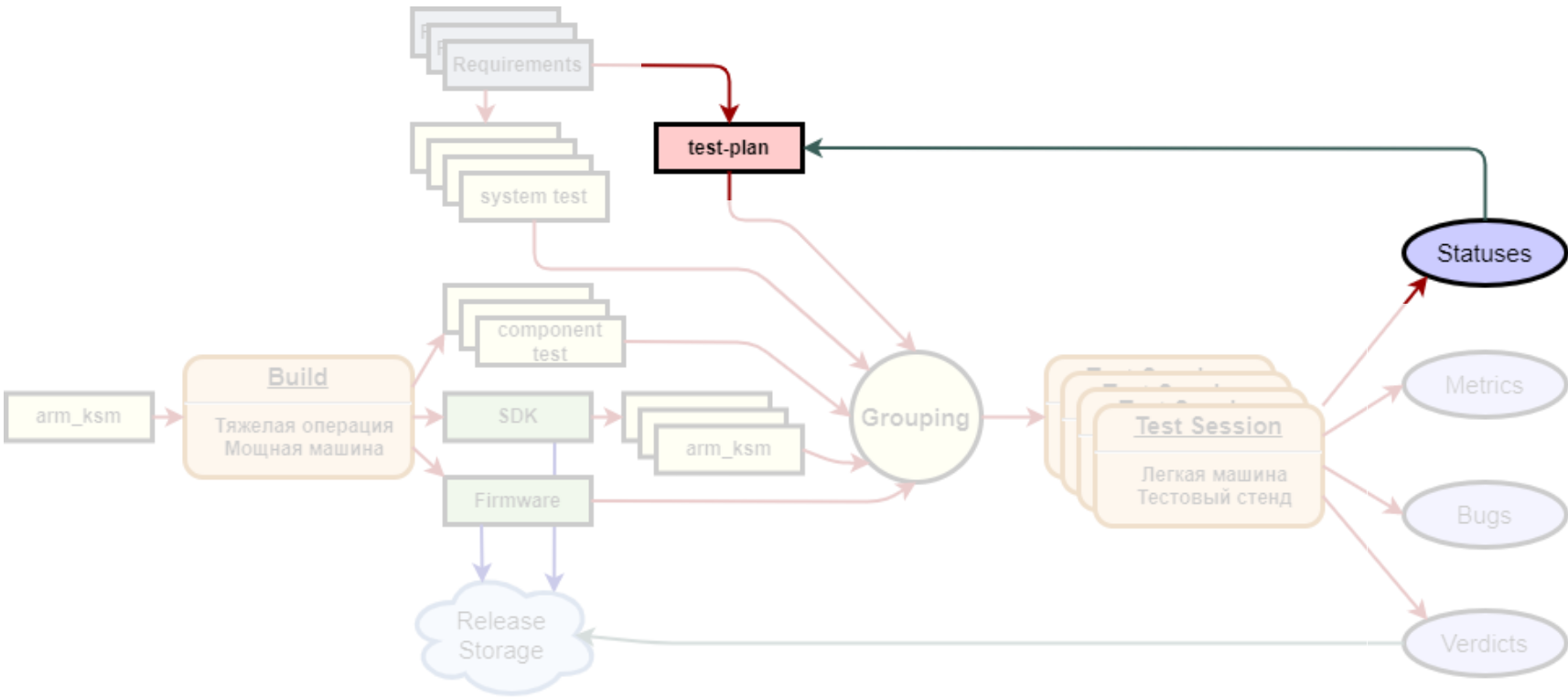
Примерная схема сборочного конвейера



Примерная схема сборочного конвейера



Примерная схема сборочного конвейера



Как запустить тестовую сессию?

Необходимо распознавать все тесты

Делить их на группы

Выбирать агенты из пула имеющихся

Уметь давать задание агенту

Уметь публиковать результаты выполнения задания

Опционально автоматически заводить баги

Как запустить тестовую сессию?

Необходимо распознавать все
тесты

Делить их на группы

Выбирать агенты из пула имеющихся

Уметь давать задание агенту

Уметь публиковать результаты
выполнения задания

Опционально автоматически
заводить баги

Как запустить тестовую сессию?

Необходимо распознавать все тесты

Делить их на группы

Выбирать агенты из пула имеющихся

Уметь давать задание агенту

Уметь публиковать результаты выполнения задания

Опционально автоматически заводить баги

Как запустить тестовую сессию?

Необходимо распознавать все тесты

Делить их на группы

Выбирать агенты из пула имеющихся

Уметь давать задание агенту

Уметь публиковать результаты выполнения задания

Опционально автоматически заводить баги

Как запустить тестовую сессию?

Необходимо распознавать все тесты

Делить их на группы

Выбирать агенты из пула имеющихся

Уметь давать задание агенту

**Уметь публиковать результаты
выполнения задания**

Опционально автоматически
заводить баги

Как запустить тестовую сессию?

Необходимо распознавать все тесты

Делить их на группы

Выбирать агенты из пула имеющихся

Уметь давать задание агенту

Уметь публиковать результаты выполнения задания

Опционально автоматически заводить баги

Что нужно делать на тестовом стенде?

**Нужно знать на каком стенде
запущена тестовая сессия**

Нужно предоставить тесту
интерфейс управления тестовыми
инструментами

Нужно подготовить тестовое
окружение

Запустить тест

Прибраться после тестовой сессии

Загрузить тестовые артефакты в
хранилище

Что нужно делать на тестовом стенде?

Нужно знать на каком стенде
запущена тестовая сессия

Нужно предоставить тесту
интерфейс управления тестовыми
инструментами

Нужно подготовить тестовое
окружение

Запустить тест

Прибраться после тестовой сессии

Загрузить тестовые артефакты в
хранилище

Что нужно делать на тестовом стенде?

Нужно знать на каком стенде
запущена тестовая сессия

Нужно предоставить тесту
интерфейс управления тестовыми
инструментами

**Нужно подготовить тестовое
окружение**

Запустить тест

Прибраться после тестовой сессии

Загрузить тестовые артефакты в
хранилище

Что нужно делать на тестовом стенде?

Нужно знать на каком стенде
запущена тестовая сессия

Нужно предоставить тесту
интерфейс управления тестовыми
инструментами

Нужно подготовить тестовое
окружение

Запустить тест

Прибраться после тестовой сессии

Загрузить тестовые артефакты в
хранилище

Что нужно делать на тестовом стенде?

Нужно знать на каком стенде
запущена тестовая сессия

Нужно предоставить тесту
интерфейс управления тестовыми
инструментами

Нужно подготовить тестовое
окружение

Запустить тест

Прибраться после тестовой сессии

Загрузить тестовые артефакты в
хранилище

Что нужно делать на тестовом стенде?

Нужно знать на каком стенде
запущена тестовая сессия

Нужно предоставить тесту
интерфейс управления тестовыми
инструментами

Нужно подготовить тестовое
окружение

Запустить тест

Прибраться после тестовой сессии

**Загрузить тестовые артефакты в
хранилище**

Что говорит мировая практика?

**Подходящих инструментов в
открытом доступе нет**

Код работы с аппаратными
инструментами придется писать
самим

Есть хорошие тестовые движки

Что говорит мировая практика?

Подходящих инструментов в
открытом доступе нет

**Код работы с аппаратными
инструментами придется писать
самим**

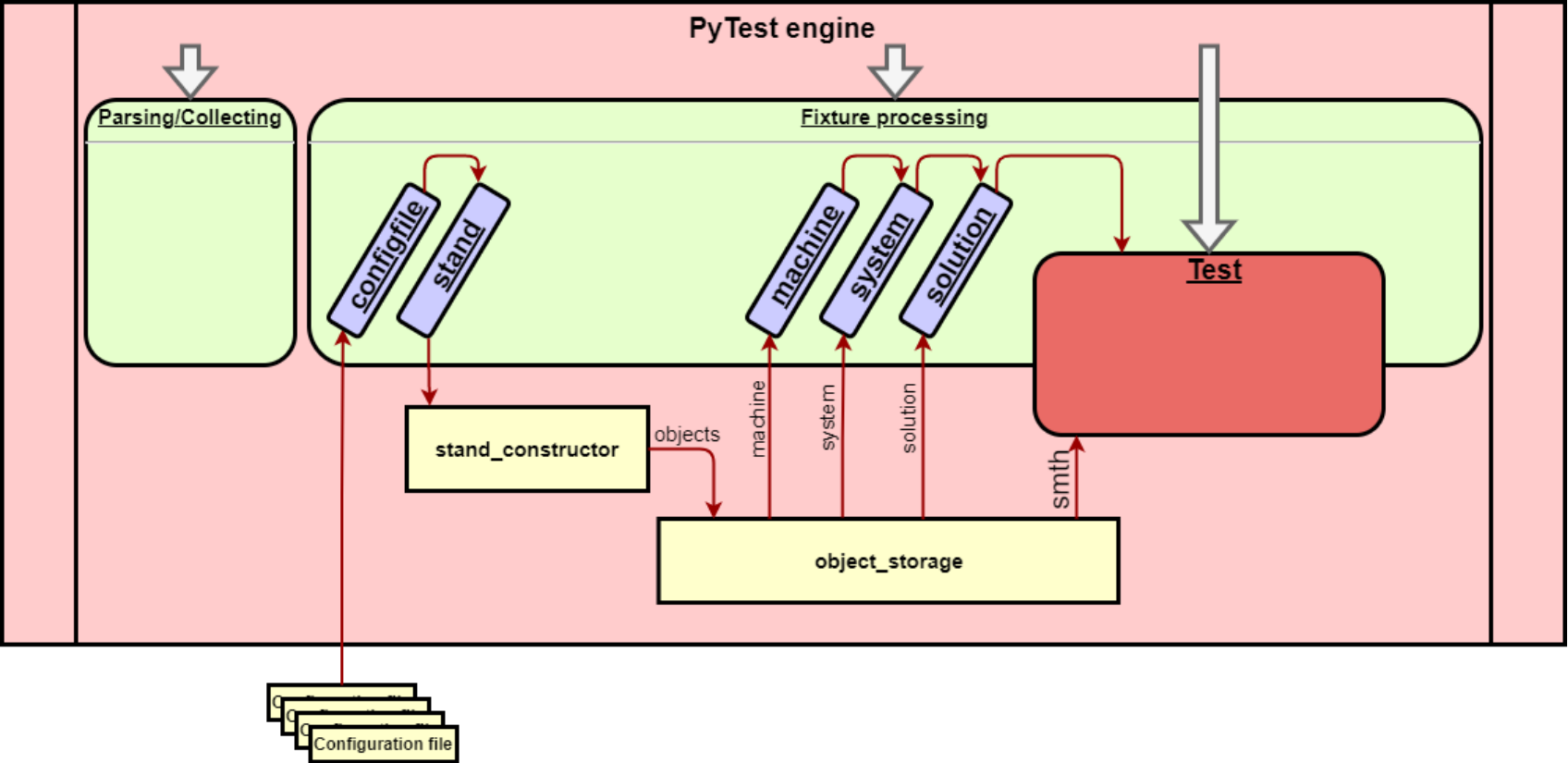
Есть хорошие тестовые движки

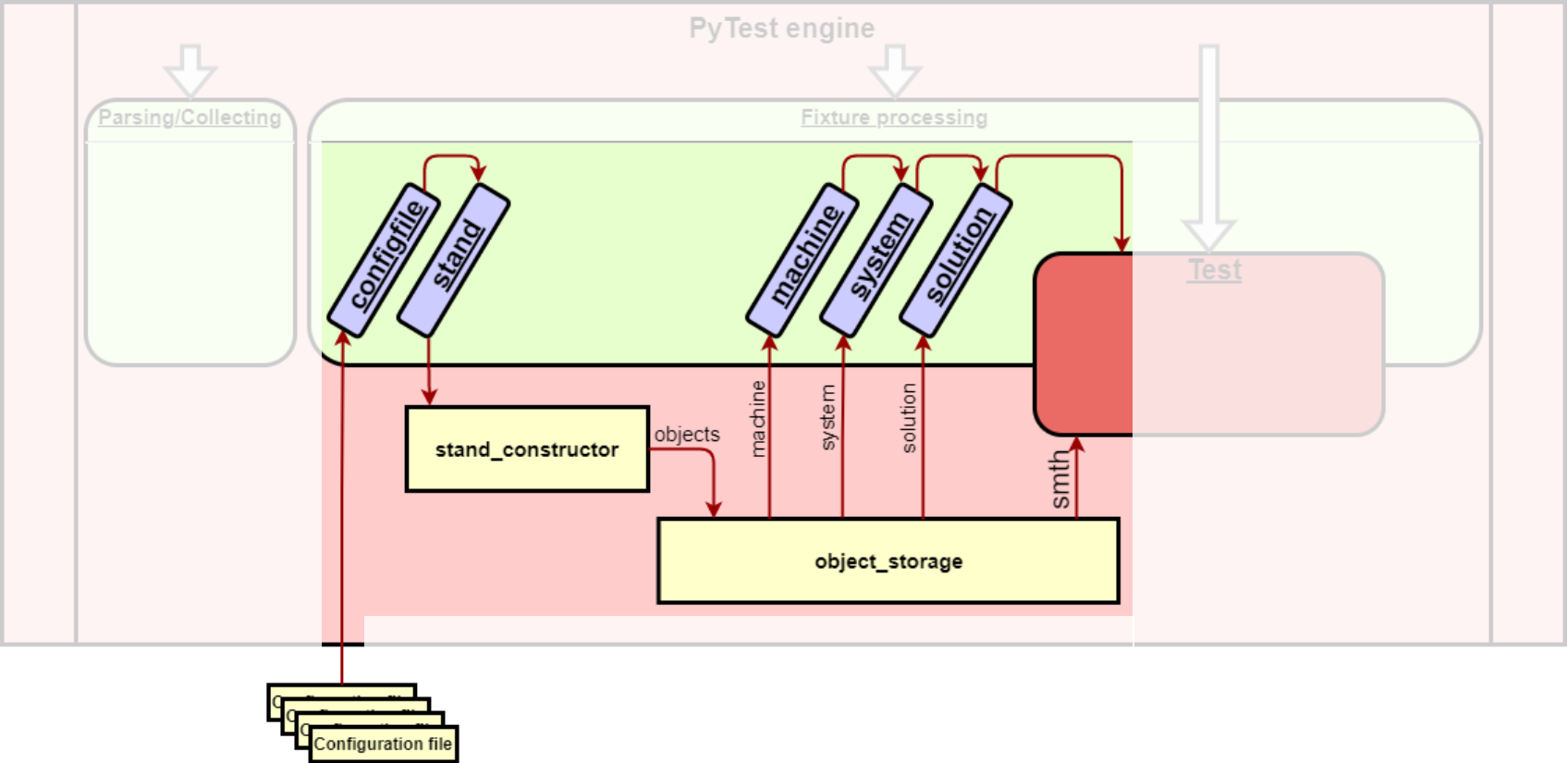
Что говорит мировая практика?

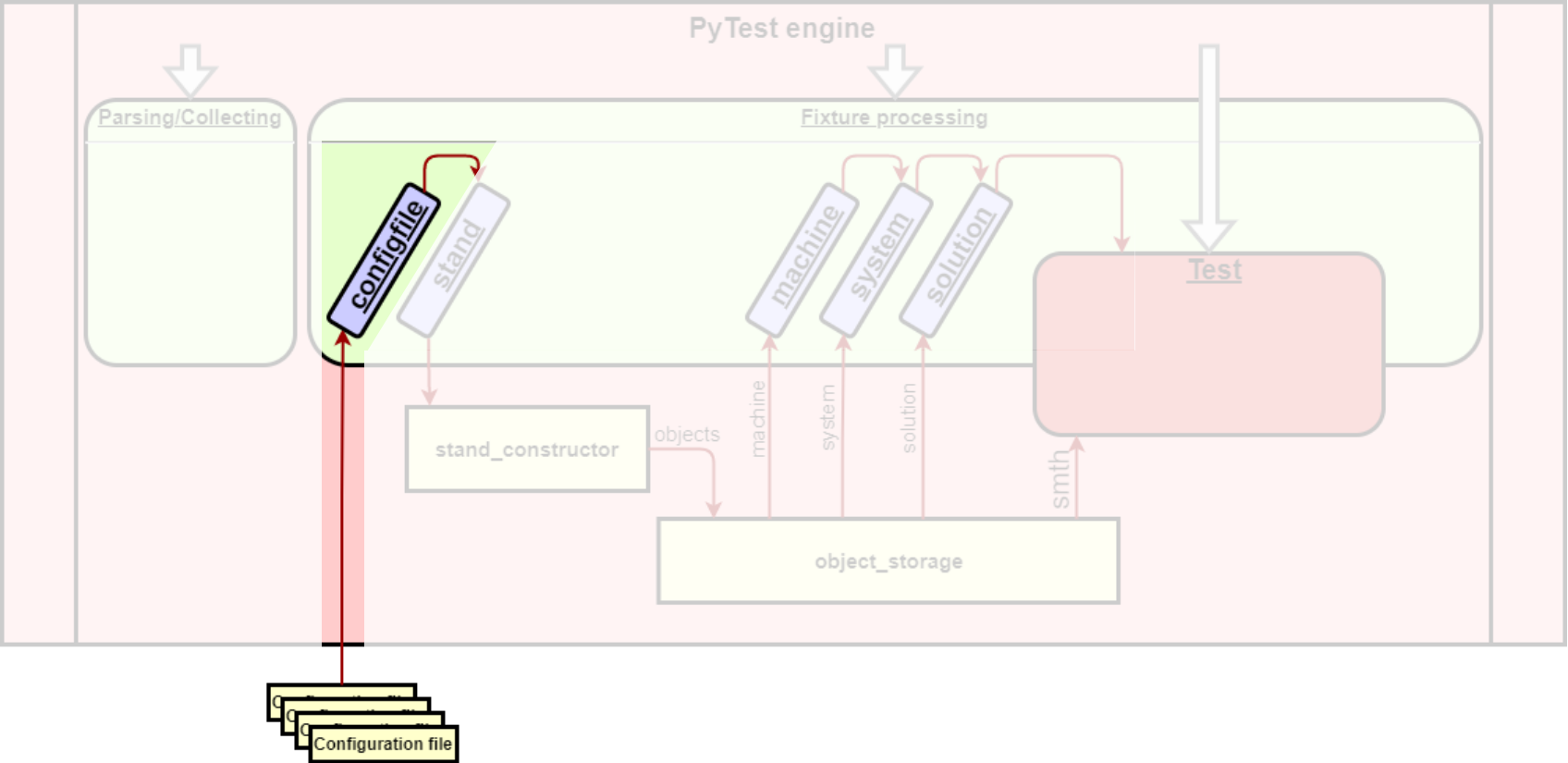
Подходящих инструментов в
открытом доступе нет

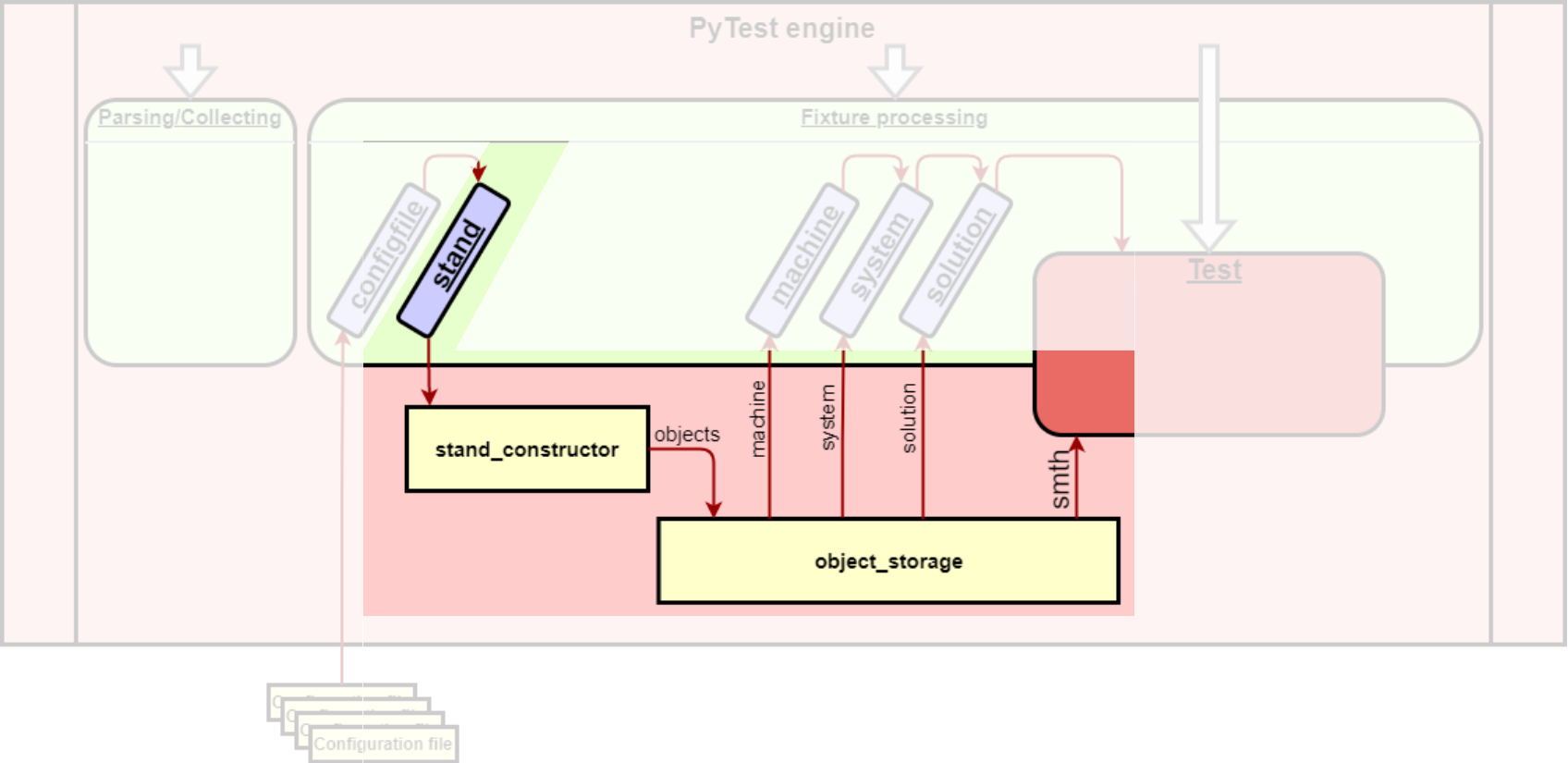
Код работы с аппаратными
инструментами придется писать
самим

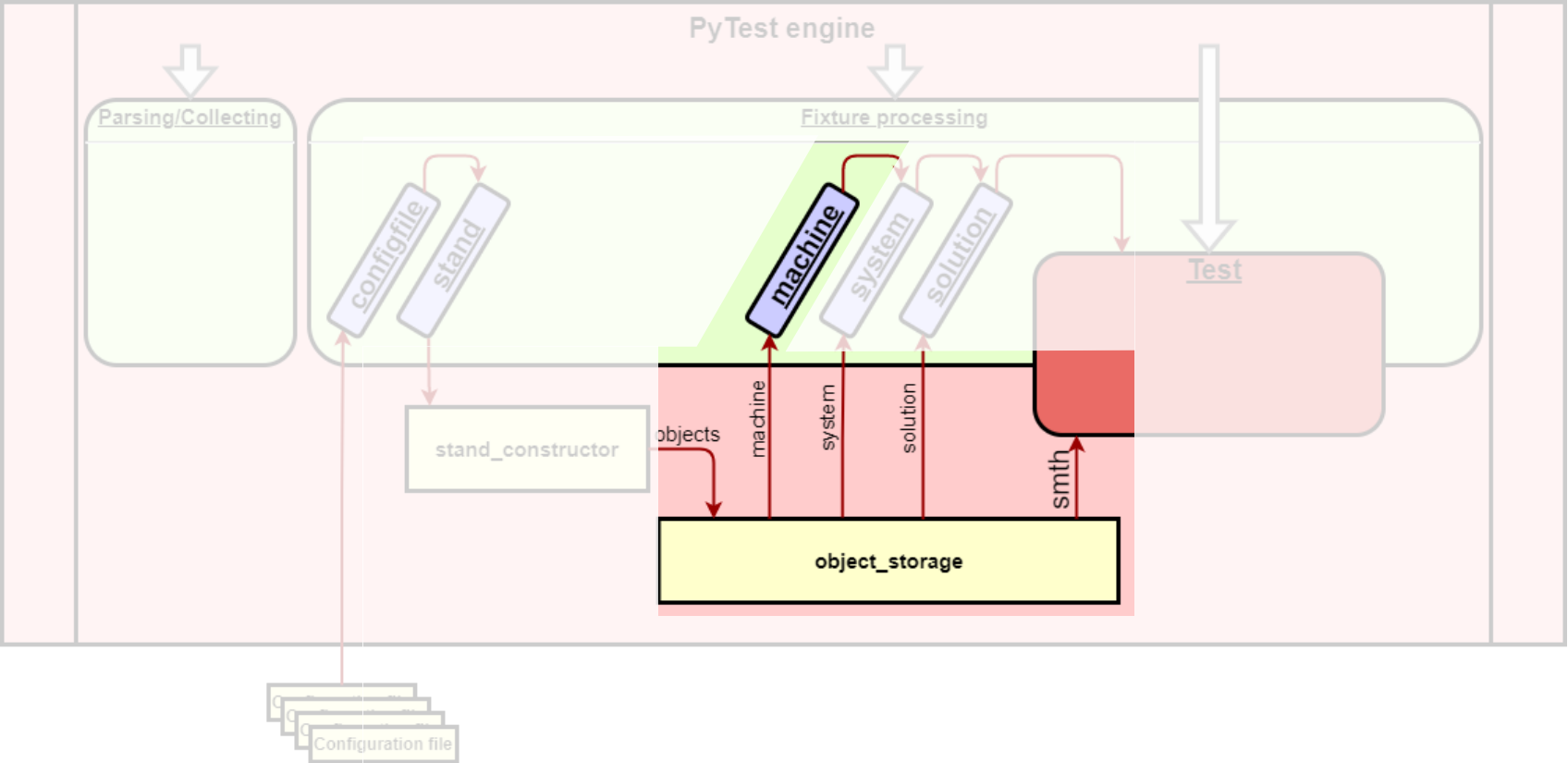
Есть хорошие тестовые движки

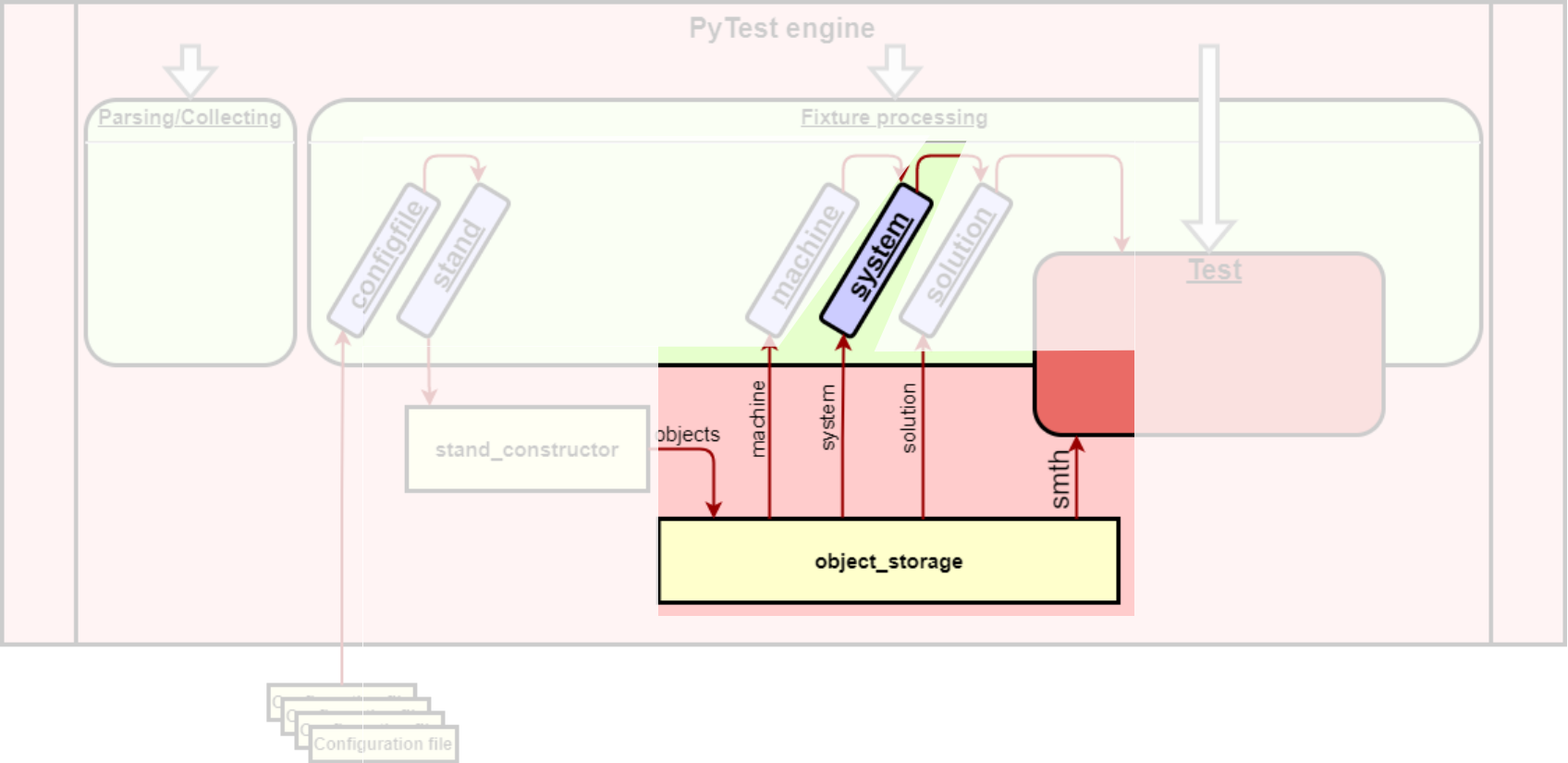


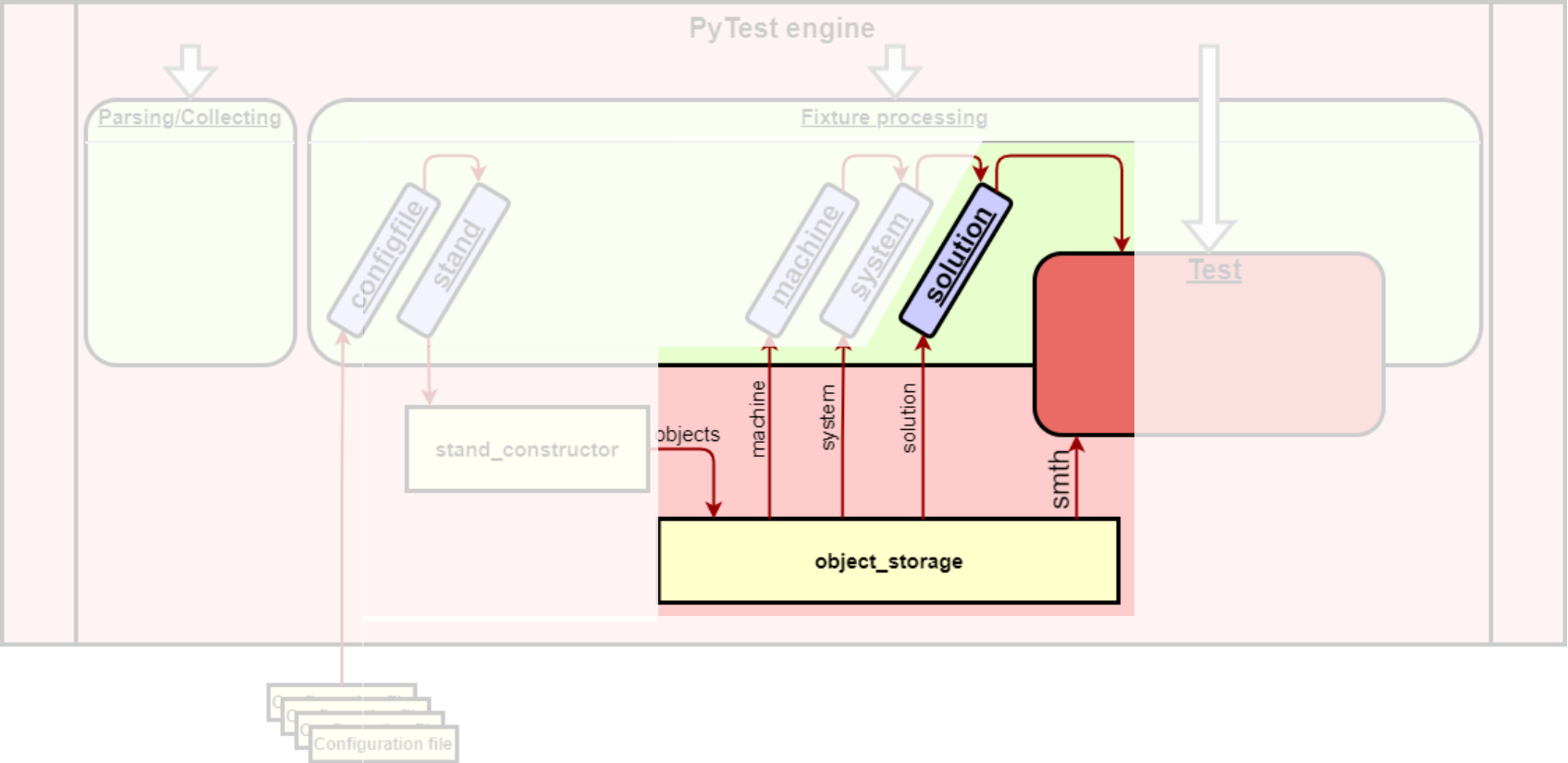












Фикстура machines

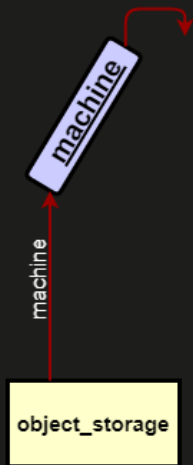
62

```
@pytest.fixture()
def machines(request, stand):
    """
    Fixture gets all machines from stand container.
    Scope: function.
    """

    def teardown():
        try:
            for machine_name, machine in stand.get_elements_by_type(Machine).items():
                machine.stop()
        except Exception as e:
            raise KTTMachineError("Failed to teardown machine(s)!", error=e)

    request.addfinalizer(teardown)

    try:
        return stand.get_elements_by_type(Machine)
    except Exception as e:
        raise KTTMachineError("Failed to setup machine(s)!", error=e)
```



Фикстура systems

63

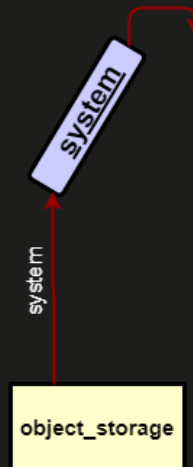
```
@pytest.fixture()
def systems(request, stand, machines):
    """
    Fixture gets all systems from stand container and deploy them to the associated machines.
    Scope: function.
    """

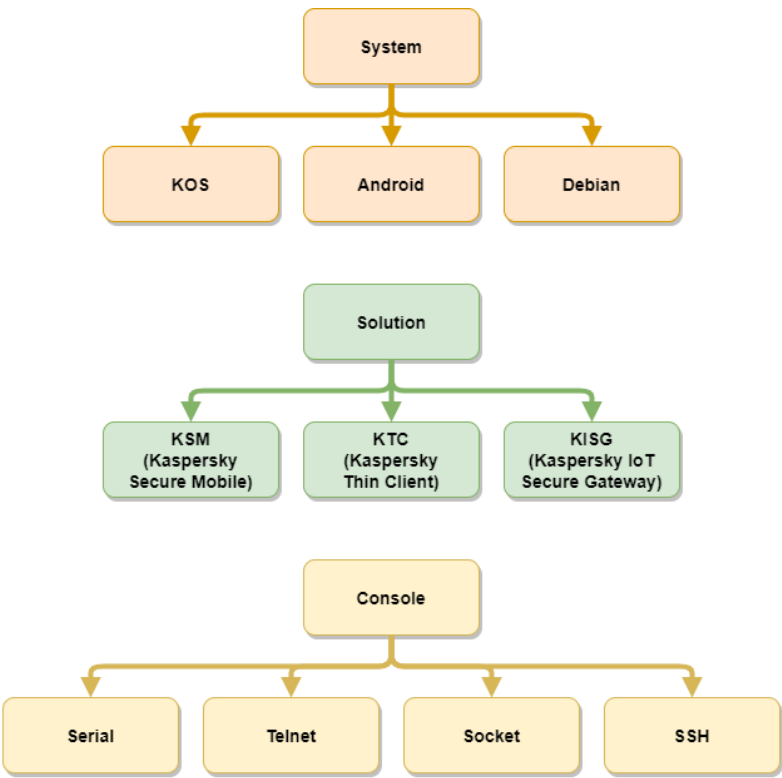
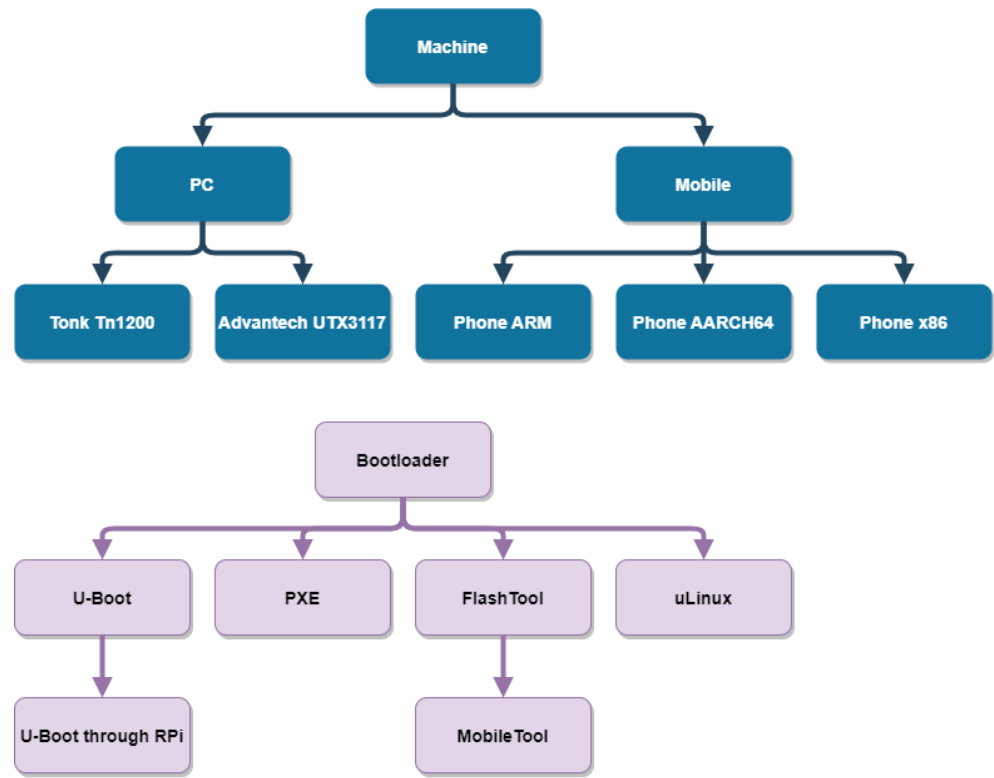
    def teardown():
        try:
            systems = stand.get_elements_by_type(OperationalSystem)
            for system in systems:
                systems[system].off()
        except Exception as e:
            raise KTTSystemError("Failed to teardown system(s)!", error=e)

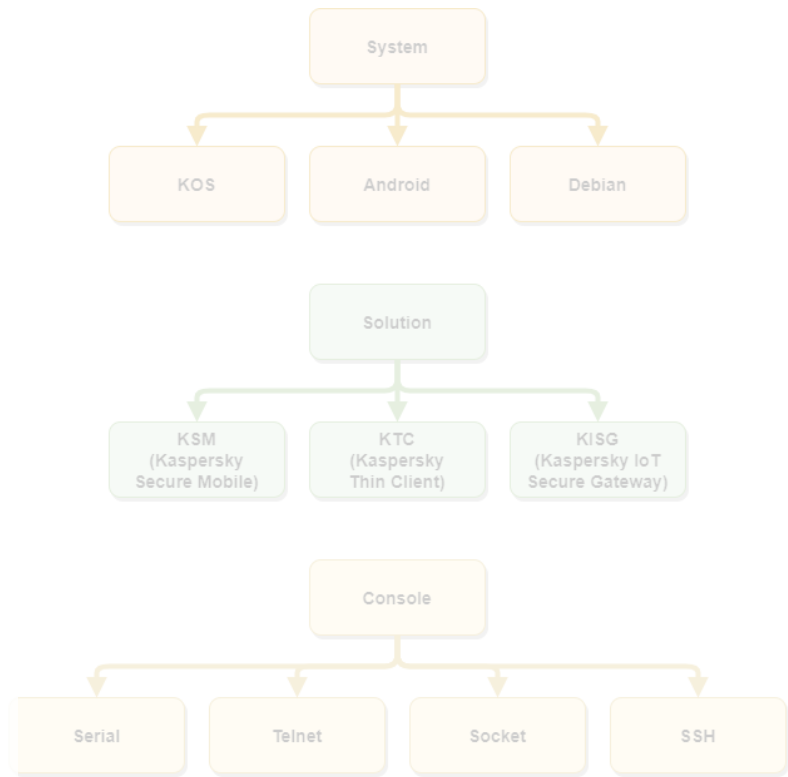
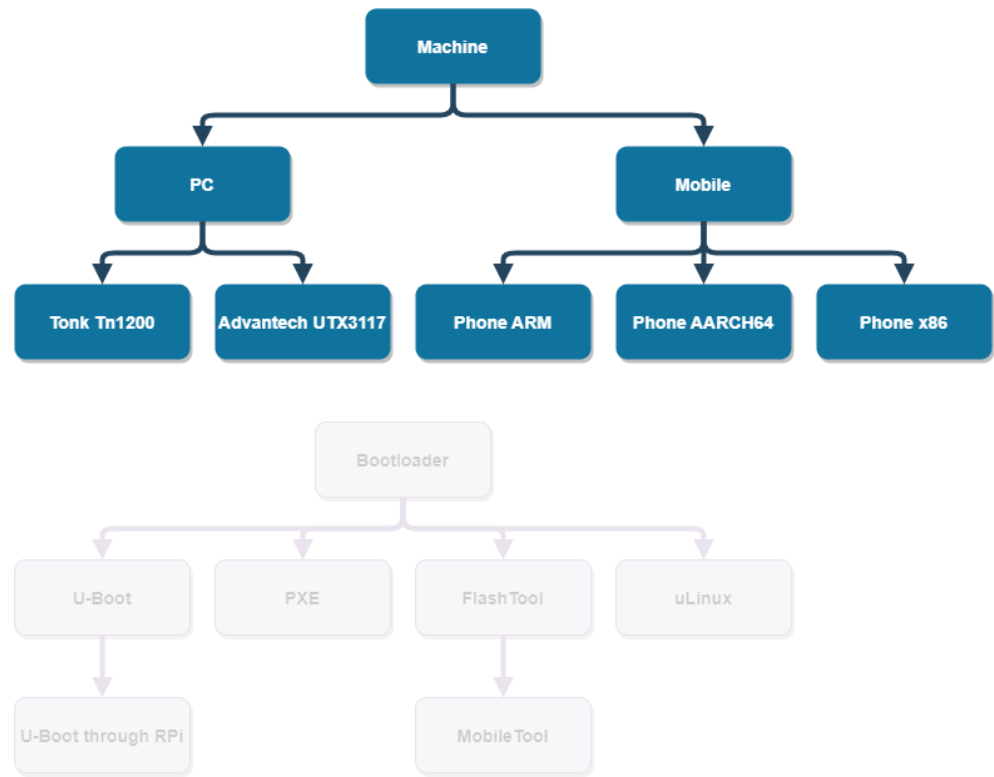
    request.addfinalizer(teardown)

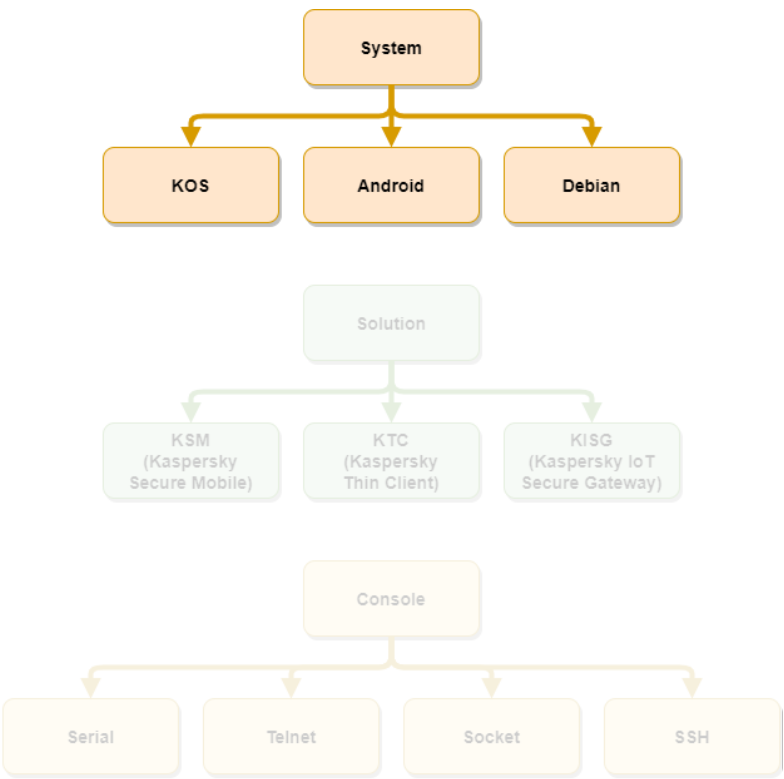
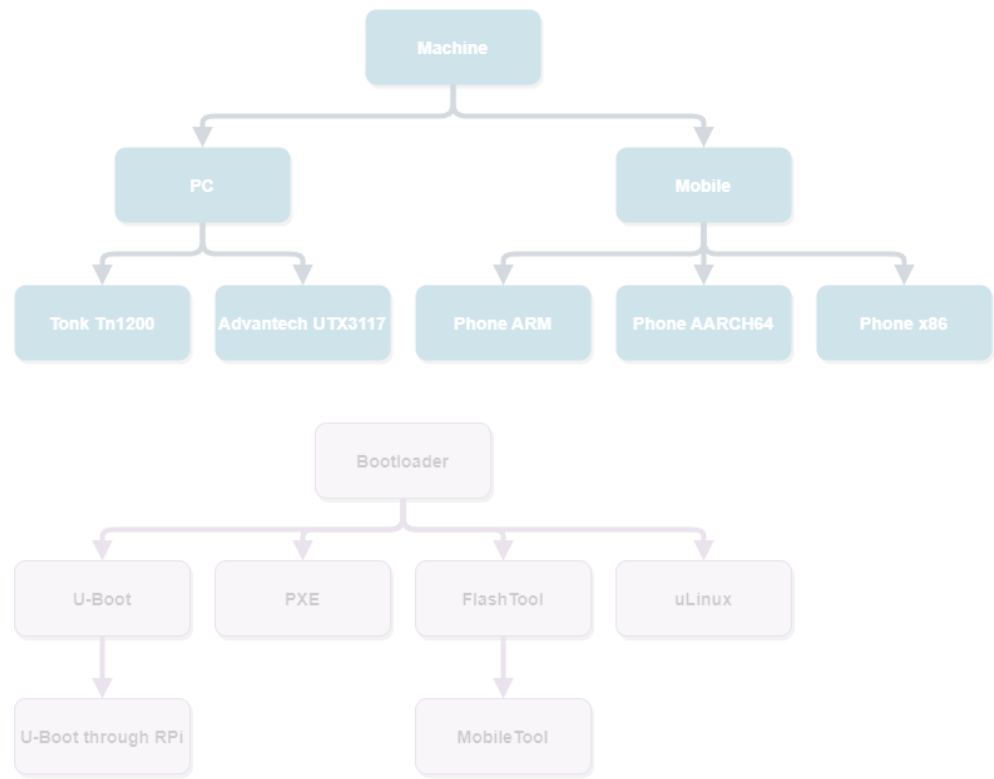
    try:
        from ktt.libs.system import OperationalSystem
        systems = stand.get_elements_by_type(OperationalSystem)

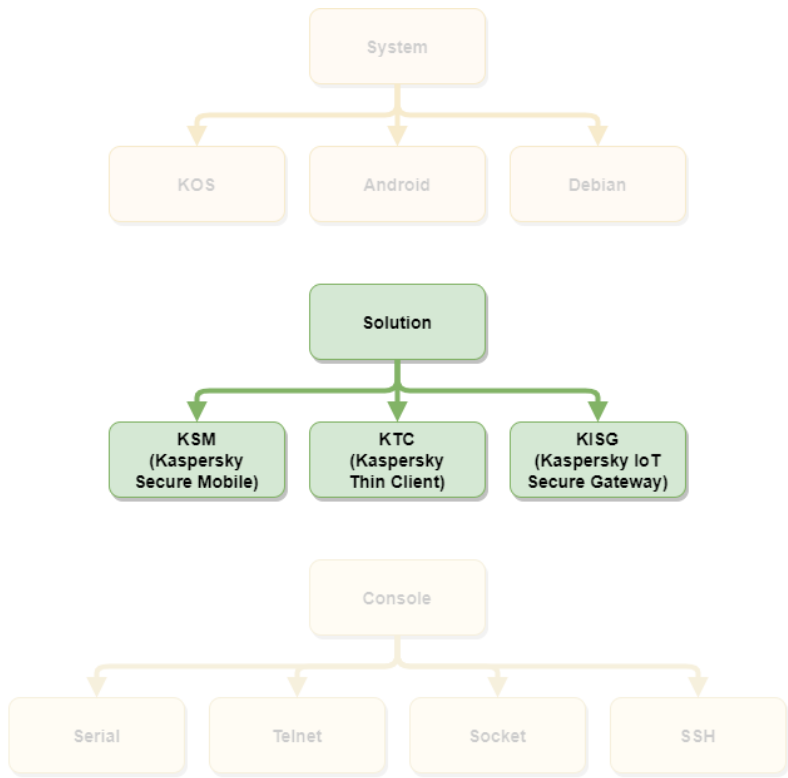
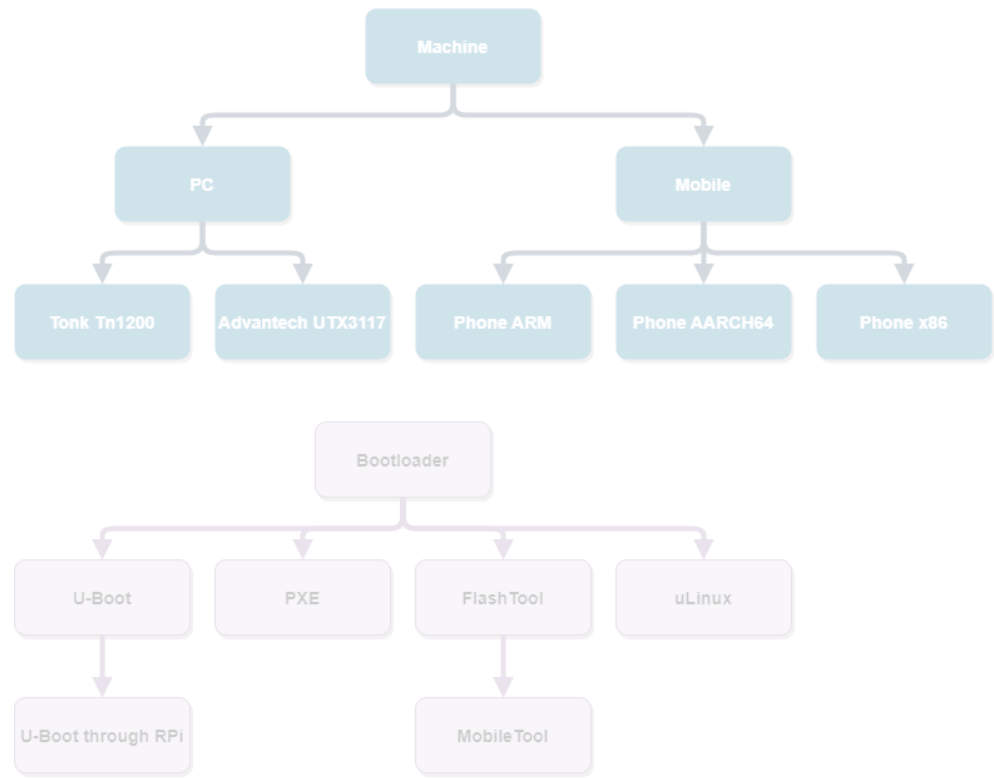
        for machine_name, machine in machines.items():
            machine.deploy()
            machine.boot()
        return systems
    except Exception as e:
        raise KTTSystemError("Failed to setup 'system(s)!", error=e)
```

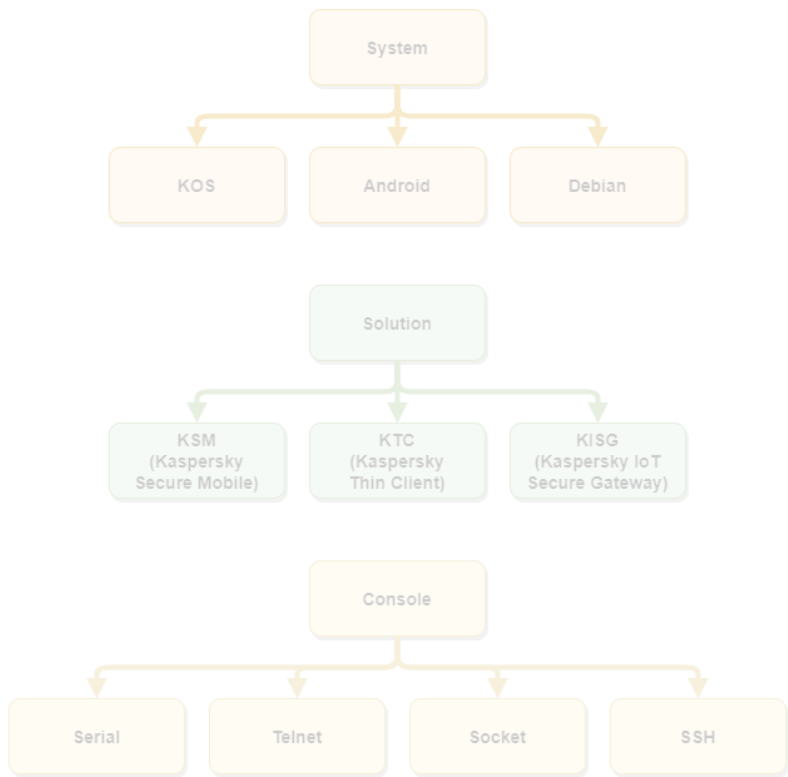
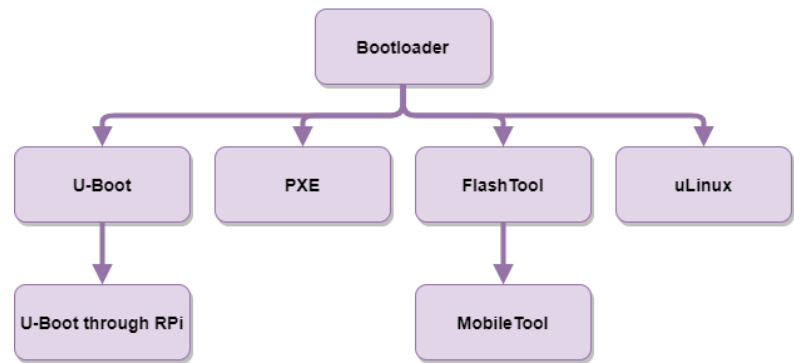
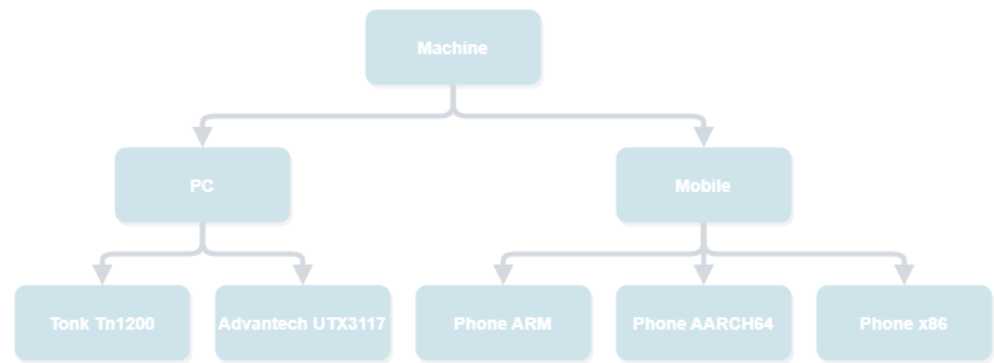


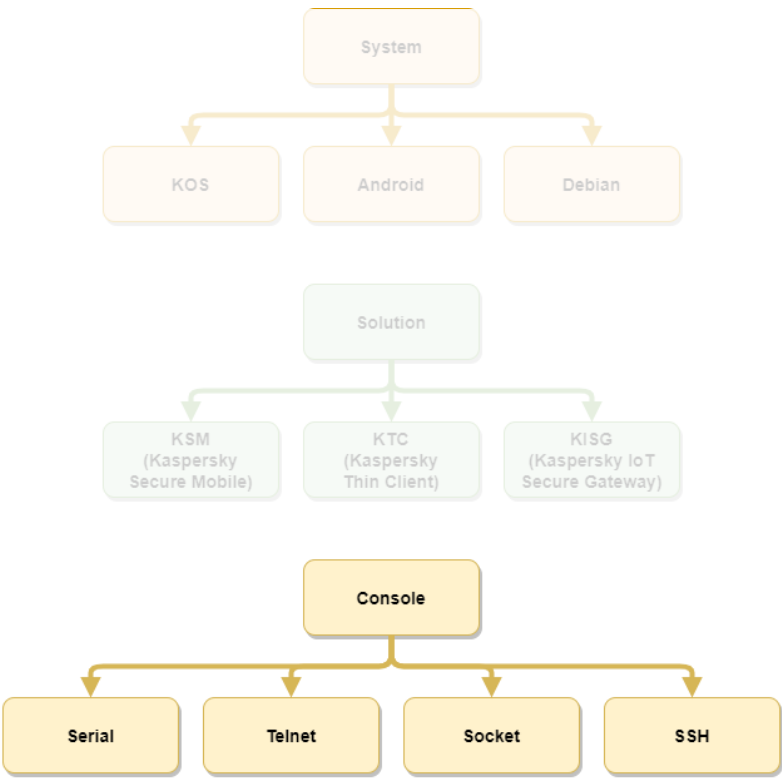
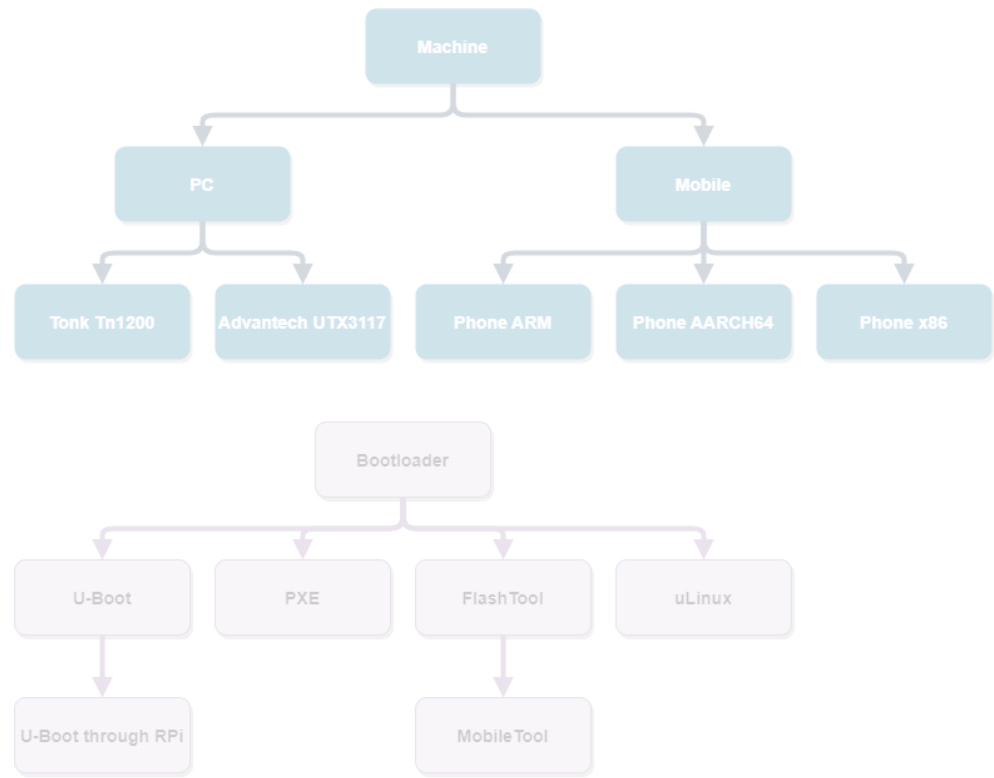


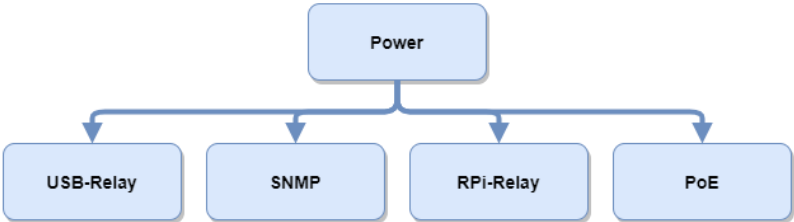
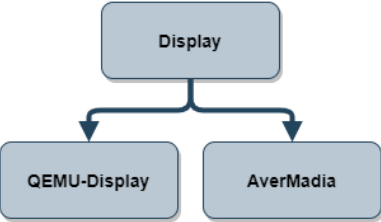
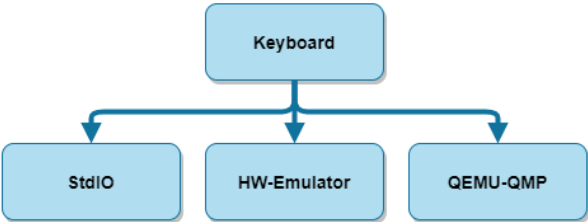


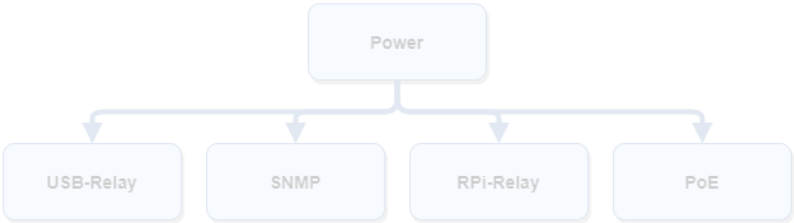
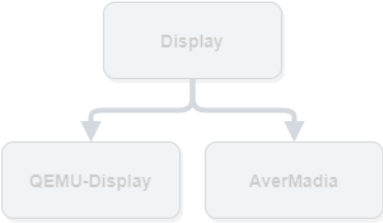
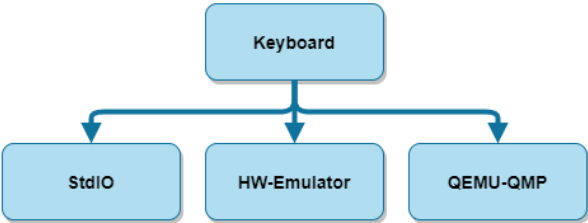


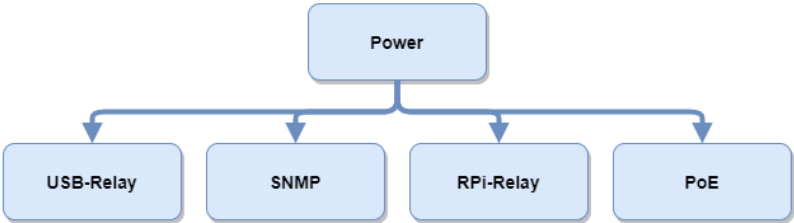
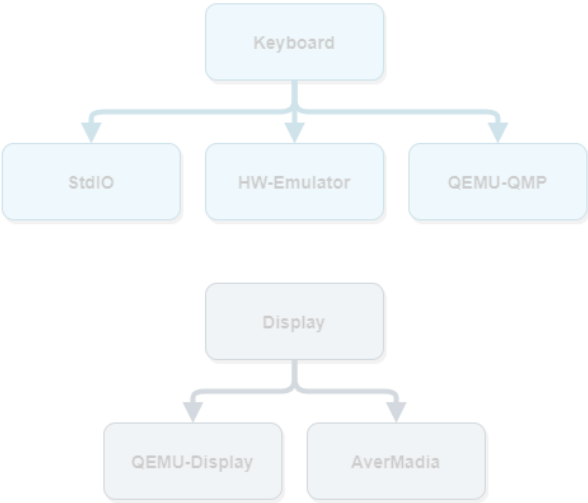


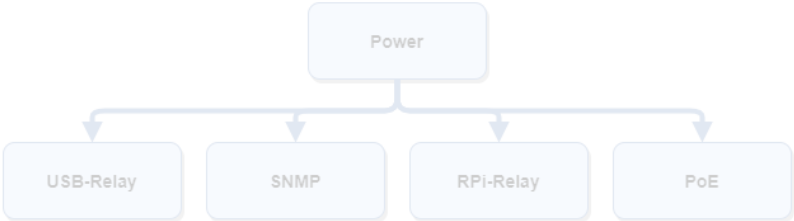
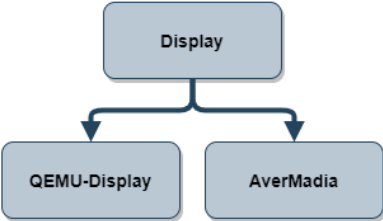
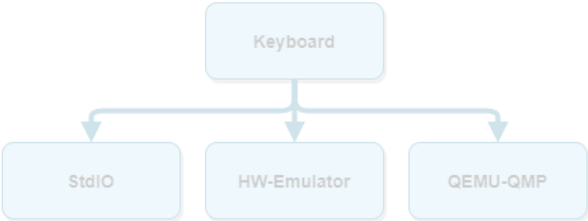












Драйвера тестовых инструментов

Предоставляем тесту набор объектов, являющийся интерфейсом к программным и аппаратным ресурсам стенда

Настраиваем набор с помощью конфигурационных файлов

Конфигурационные файлы можем выбирать в зависимости от...

Драйвера тестовых инструментов

Предоставляем тесту набор объектов, являющийся интерфейсом к программным и аппаратным ресурсам стенда

Настраиваем набор с помощью конфигурационных файлов

Конфигурационные файлы можем выбирать в зависимости от...

Драйвера тестовых инструментов

Предоставляем тесту набор объектов, являющийся интерфейсом к программным и аппаратным ресурсам стенда

Настраиваем набор с помощью конфигурационных файлов

Конфигурационные файлы можем выбирать в зависимости от...

Набор инструментов для теста мобильной ОС

77

```
stand:
machine:
  type: phone_aarch64
power:
  type: mobile_tool
  address: '192.168.0.2'
  login: $(KSM/mobile-tool-1/username@password_manager)
  password: $(KSM/mobile-tool-1/password@password_manager)
console:
  type: telnet
  address: '192.168.0.2:6000'
bootloader:
  type: mobile_tool
  address: '192.168.0.2'
  login: $(KSM/mobile-tool-1/username@password_manager)
  password: $(KSM/mobile-tool-1/password@password_manager)
system:
  type: KOS
```

Набор инструментов для теста тонкого клиента

78

```
stand:
machine:
  type: tn1200
  power:
    type: snmp
    address: "kocherga.local.ru"
    outlet_id: 12
console:
  class: SSHConsole
  address: "192.168.0.222"
  login: $(KTC/ktc3/rpi/username@password_manager)
  password: $(KTC/ktc3/rpi/password@password_manager)
system:
  type: KOS
bootloader:
  - type: pxe
    mode: legacy
  - type: grub
    mode: legacy
    loader: multiboot
  - type: ulinux
keyboard:
  type: hardware_emulator
  class: HardwareEmulatorKeyboard
console:
  type: telnet
  address: "192.168.8.333:23"
```

Набор инструментов для теста безопасного шлюза

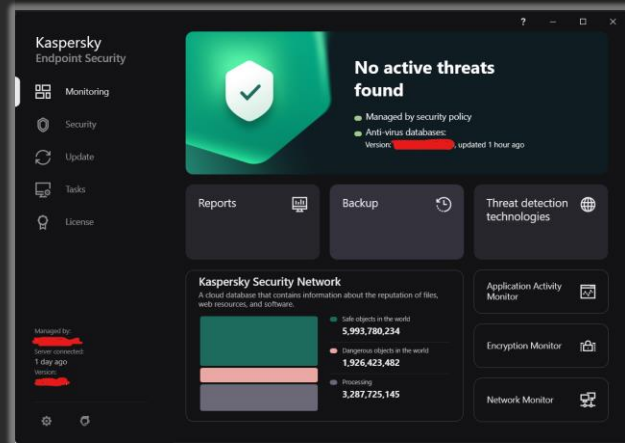
79

```
stand:
  ksc:
    url: "https://10.165.1.162:8080"
    hostname: kiskgksc01.local
  machine:
    type: utx3117
    net_interface:
      - role: wan
        address: 192.168.2.202
      - role: lan
        address: 192.168.1.1
  power:
    type: snmp
    address: "kocherga.local.ru"
    outlet_id: 20
  console:
    type: telnet
    class: TelnetConsole
    address: console1.ru:6002
  bootloader:
    - type: pxe
      mode: legacy
      loader: mboot.c32
    - type: ulinux
  system:
    type: KOS
```

Тест настройки политик доступа через KSC на linux

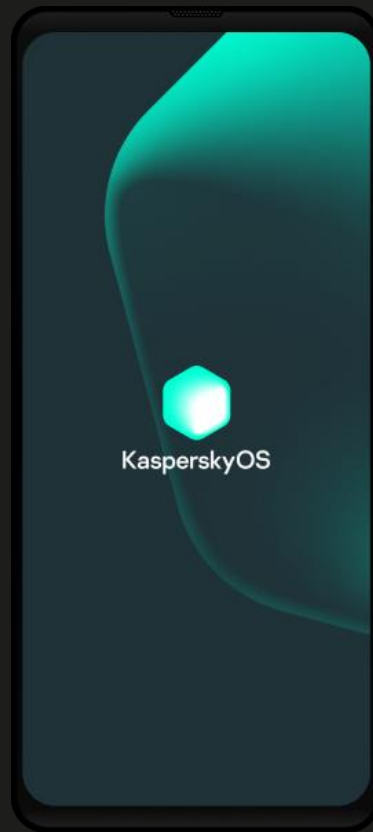
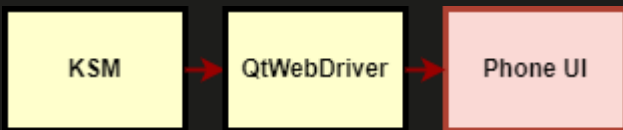
80

```
def test_policy(ksc, avp):  
    policy = 'user_policy'  
    ksc.deny_policy(policy)  
    is_enabled = avp.check_policy(policy)  
    assert is_enabled, f'Policy {policy} has not been set'  
    ksc.allow_policy(policy)  
    is_enabled = avp.check_policy(policy)  
    assert not is_enabled, f'Policy {policy} has not been cleared'
```



Тест настройки политик доступа через KSC на мобильном телефоне

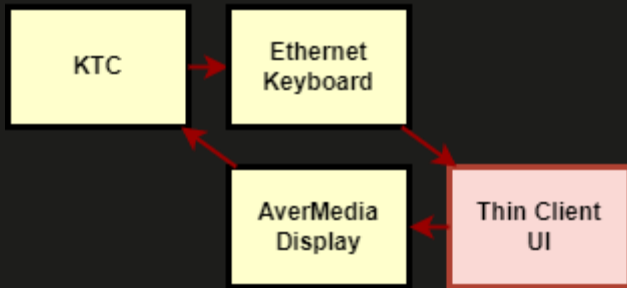
```
def test_policy(ksc, ksm):  
    policy = 'user_policy'  
    ksc.deny_policy(policy)  
    is_enabled = ksm.check_policy(policy)  
    assert not is_enabled, f'Policy {policy} has not been set'  
    ksc.allow_policy(policy)  
    is_enabled = ksm.check_policy(policy)  
    assert is_enabled, f'Policy {policy} has not been cleared'
```



Тест настройки политик доступа через KSC на тонком клиенте

82

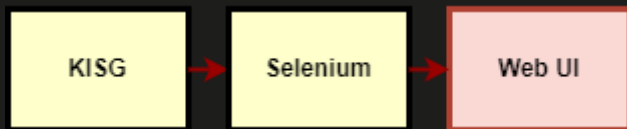
```
def test_policy(ksc, ktc):  
    policy = 'user_policy'  
    ksc.deny_policy(policy)  
    is_enabled = ktc.check_policy(policy)  
    assert not is_enabled, f'Policy {policy} has not been set'  
    ksc.allow_policy(policy)  
    is_enabled = ktc.check_policy(policy)  
    assert is_enabled, f'Policy {policy} has not been cleared'
```



Тест настройки политик доступа через KSC на безопасном шлюзе

83

```
def test_policy(ksc, ksig):  
    policy = 'user_policy'  
    ksc.deny_policy(policy)  
    is_enabled = ksig.check_policy(policy)  
    assert not is_enabled, f'Policy {policy} has not been set'  
    ksc.allow_policy(policy)  
    is_enabled = ksig.check_policy(policy)  
    assert is_enabled, f'Policy {policy} has not been cleared'
```



Выводы

Существенно облегчили разработку системных тестов KasperskyOS

Обеспечили легкое добавление новых аппаратных стендов

Разработали единый инструмент, который позволяет разрабатывать и запускать тесты как локально, так и на конвейере

Тесты теперь могут быть кроссплатформенными

Выводы

Существенно облегчили разработку системных тестов KasperskyOS

Обеспечили легкое добавление новых аппаратных стендов

Разработали единый инструмент, который позволяет разрабатывать и запускать тесты как локально, так и на конвейере

Тесты теперь могут быть кроссплатформенными

Выводы

Существенно облегчили разработку системных тестов KasperskyOS

Обеспечили легкое добавление новых аппаратных стендов

Разработали единый инструмент, который позволяет разрабатывать и запускать тесты как локально, так и на конвейере

Тесты теперь могут быть кроссплатформенными

Выводы

Существенно облегчили разработку системных тестов KasperskyOS

Обеспечили легкое добавление новых аппаратных стендов

Разработали единый инструмент, который позволяет разрабатывать и запускать тесты как локально, так и на конвейере

Тесты теперь могут быть кроссплатформенными

Выводы

Обеспечили прогон тестов по требованиям и трассируемость результатов

Повысили качество ОС и продуктов на ее основе

Выводы

Обеспечили прогон тестов по требованиям и трассируемость результатов

Повысили качество ОС и продуктов на ее основе

Приходите
на meetup
для SDET'ов



Спасибо

kaspersky