

Embedded Devices GUI Testing with Squish Using a Board Farm Cloud

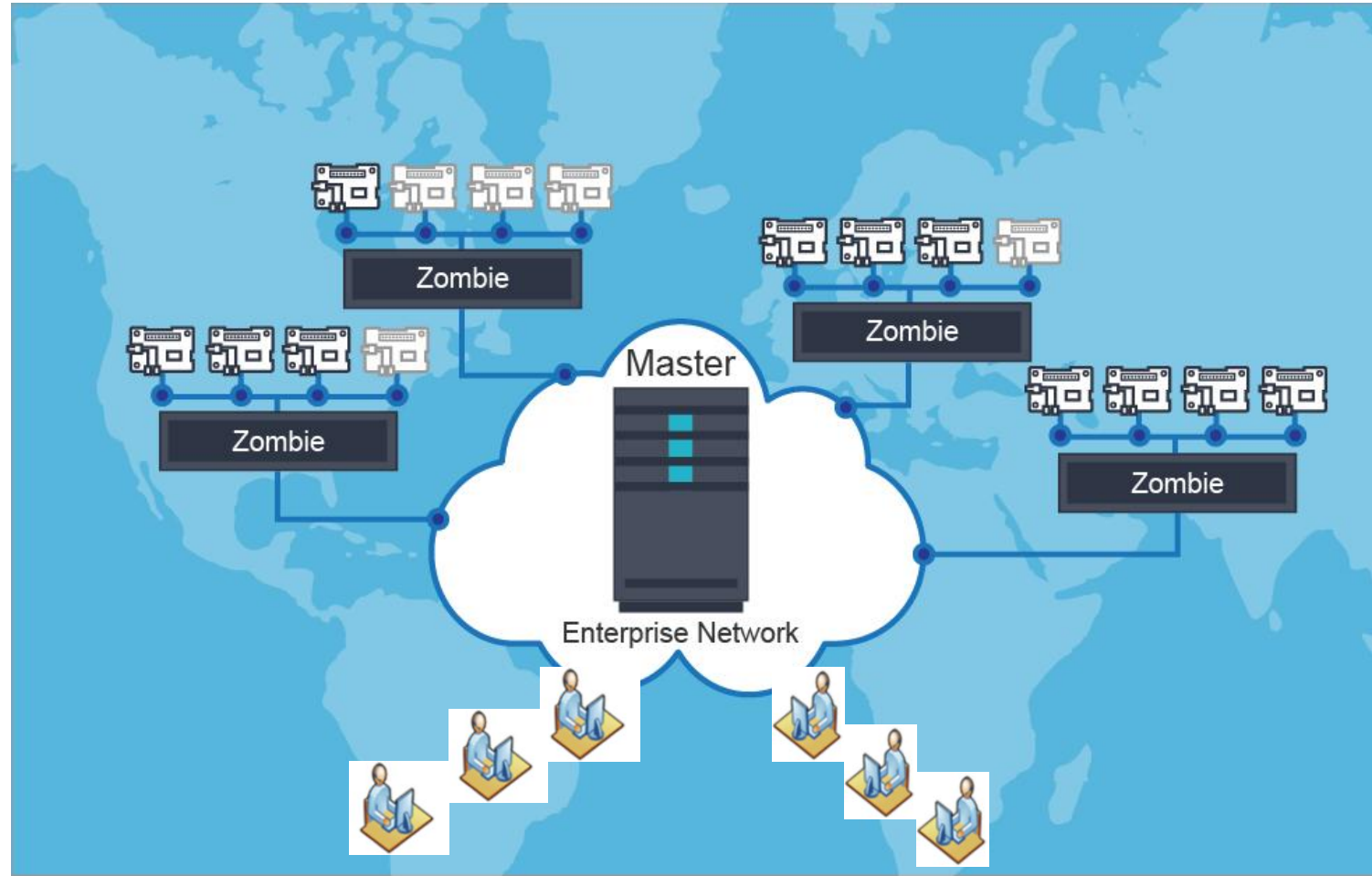
**Squish Days Europe
18th October 2019**

**Harish Bansal
harish.bansal@timesys.com**

Why build a Board Farm?

- **Better utilization of hardware resources**
 - Scarce hardware — prototypes
 - Expensive hardware
- **Reduce cycle time for new product development**
 - Distributed teams across (different sites, buildings/floors/labs, companies) can access hardware on demand and perform manual debugging and testing
 - Use the same infrastructure for continuous/automated testing
- **Reduce response time to reproduce issues**
 - Easy and on-demand access to hardware resources for reproducing field and development issues
 - Need access only once or for short durations
 - Often unable to reproduce problems on all boards
- **Typical solution: Ship, Track and Manage**
 - Low utilization of hardware resources and increase cycle time
- **Desired solution: Centralized hardware access with a few exceptions**

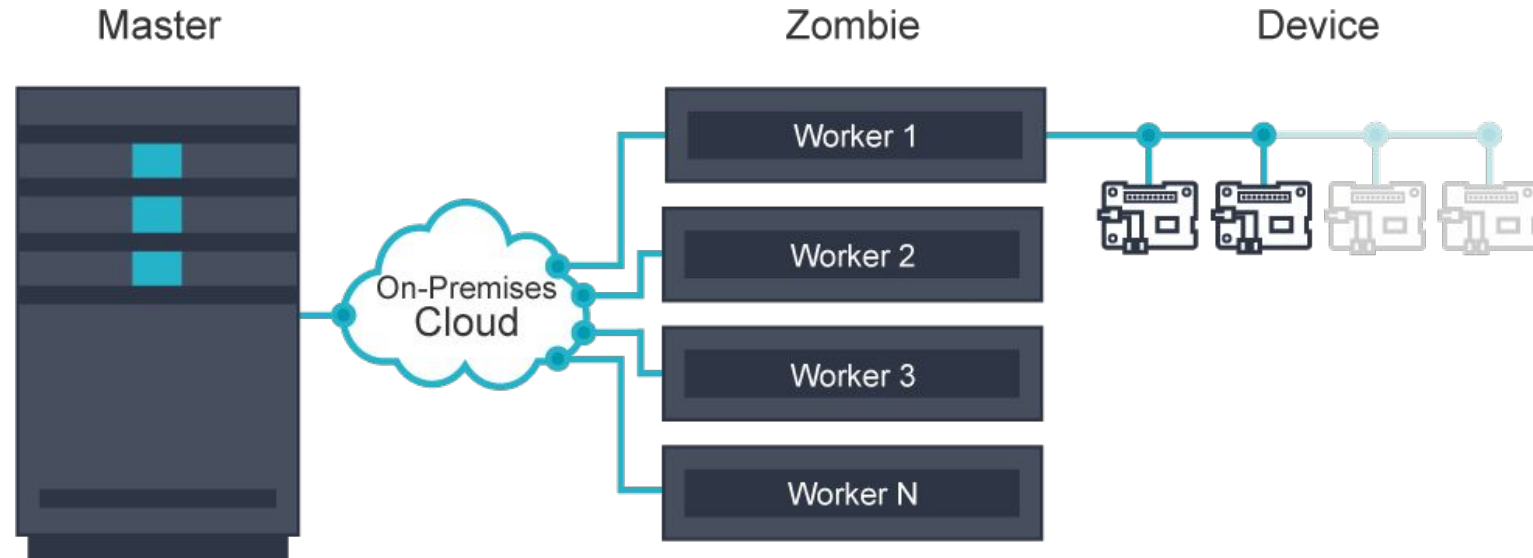
On-Premise Board Farm Cloud (BFC)



BF Dashboard
Board Management and Access

BFC Master

- **Centralized Device Management**
- **Multi-user**
- **Docker container**
- **Console access**
- **Power control**
- **Image and File transfer**
- **Hotplug Control**
- **Built-in App/Test server**



Zombie

- **Zombie (red)**
- **App/Test Server (blue)**



IO-CX

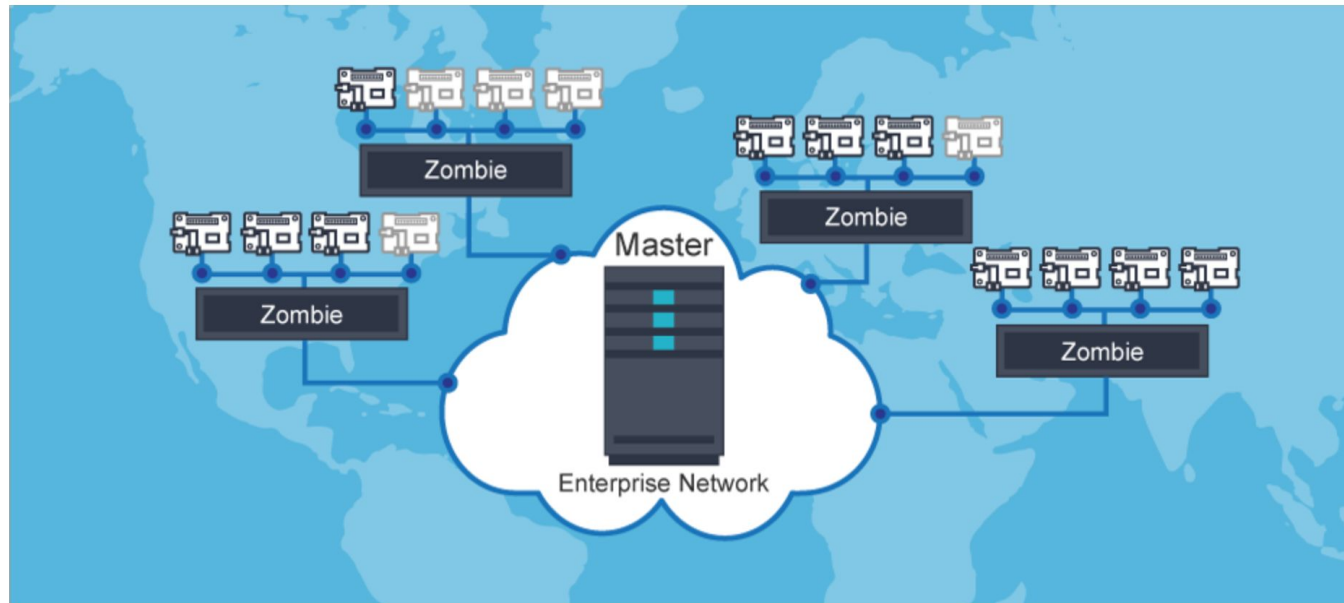
- **USB hotplugs**
- **Ethernet hotplugs**
- **SDMUX**
- **I2C bus connector**
- **GPIO connector**

Dashboard — Board Management

Timesys Zombies Devices Tests Scheduler Results Help admin

Timesys Board Farm Cloud and Test Automation Solution

Welcome to Timesys BFC and TAS powered by LAVA.



- User account management
- Zombie management
- Board management
 - Board addition/ configuration/deletion
 - Board (devices) status
 - Board allocation/ deallocation

Get Started

Add Zombie

Add Device

Devices

All Devices

My Devices

Devices

- Sign in
- Visit All Devices
- Allocate the device
- Launch Console
- Retire Device
- Visit My Devices

All Devices

[All Devices](#)
[Active Devices](#)
[My Devices](#)

Actions

[Launch Side By Side Console](#)

Show entries Search

| Device Name | Zombie Worker | Device type | Device status | Assigned To | tags | IOCX Connected? | Actions |
|-----------------------|---------------|------------------------|---------------|---------------------|------|-----------------|---------|
| am3354_respirionics-1 | BFZombie1 | am335x | Idle | — | | No | |
| am3517_evm-1 | BFZombie1 | am35XX | Idle | — | | No | |
| am437x_evm-1 | BFZombie2 | am437x | Idle | patrick.mochel | | No | |
| am572x_evm-1 | BFZombie7 | am572x_EVM | Retired | — | | — | |
| am572x_hdmi_display | BFZombie7 | external_display | Idle | prasanth.rameshbabu | | No | |
| amcc405gp-1 | BFZombie2 | PowerPC | Idle | — | | No | |
| amcc460ex-1 | BFZombie2 | PowerPC | Idle | — | | No | |
| android_raspi3-1 | BFZombie3 | RasPi3 | Idle | admin | | Yes | |
| ARK_1124H-1 | BFZombie2 | intel_atom_e3940_q_SoC | Idle | — | | No | |
| at91sama5d36_ek-1 | BFZombie7 | sama5d36ek | Idle | — | | No | |
| beaglebone-1 | BFZombie8 | beaglebone | Idle | vignesh.r | | No | |
| beaglebone_black-1 | BFZombie7 | beaglebone-black | Idle | — | | No | |
| beaglebone_black_ch_1 | Zombie1_CH | beaglebone-black | Idle | vijayan.v | | No | |

BFC-TAS / My Devices

My Devices

[All Devices](#)
[Active Devices](#)
[My Devices](#)

Actions

[Launch Side By Side Console](#)
[Refresh IOCX Status](#)

Show entries Search

| Device Name | Zombie Worker | Device type | Device status | tags | IOCX Connected | Actions |
|------------------------|---------------|-----------------|---------------|------|----------------|---------|
| Honeywell-imx6sx-phx-2 | BFZombie2 | hw_custom_imx6 | Idle | | False | |
| imx51_evk-1 | BFZombie2 | imx51 | Idle | | False | |
| IMX6Q-SABRELITE-CH | TimesysZombie | imx6q-sabrelite | Idle | | False | |
| ingenu_arm-1 | BFZombie8 | Ingenu_ARM | Idle | | False | |

Console

- Device must be allocated to access this page
 - **Power Control**
 - Green = ON
 - Red = OFF
 - **New Console Session**
 - Serial
 - SSH
 - ADB (Android)
 - **Video & Audio Streaming**
 - **IO-CX Menu**
 - Green = controlled by device
 - Red = controlled by zombie
 - **Download Log**
 - **Image Transfer**
 - **Release Device**

BFC-TAS / Device Rpi3_Nav (DUT1) / Console Rpi3_Nav

Console

IO-CX | **SDMUX: device** | Hotplug1 | Hotplug2 | Hotplug3 | Hotplug4

Controls

- Console Session ▾
 - Serial
 - SSH
 - ADB
- Video Streaming
- Audio Streaming
- Image Browser
- Download Log
- Power
- SDMux
- Network Boot
- Release Device
- Info

```

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.111.3 netmask 255.255.255.0 broadcast 192.168.111.255
      inet6 fe80::3f8:8210:470b:5a3f prefixlen 64 scopeid 0x20<link>
      ether b8:27:eb:08:fb:52 txqueuelen 1000 (Ethernet)
      RX packets 23470 bytes 29523240 (28.1 MiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 14767 bytes 3701181 (3.5 MiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
      loop txqueuelen 1000 (Local Loopback)
      RX packets 6006 bytes 2996787 (2.8 MiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 6006 bytes 2996787 (2.8 MiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~$ █
  
```

[Capture Screenshot](#)

Remote Device Access

Console

IO-CX | SDMUX: device | Hotplug1 | Hotplug2 | Hotplug3 | Hotplug4

Controls

- Console Session
- Serial
- SSH
- ADB

Video Streaming

Audio Streaming

Image Browser

Download Log

Power

- off
- on
- reboot

SDMUX

Network Boot

Release Device

Info

```

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.111.3 netmask 255.255.255.0 broadcast 192.168.111.255
inet6 fe80::3f8:8210:470b:5a3f prefixlen 64 scopeid 0x20<link>
ether b8:27:eb:08:fb:52 txqueuelen 1000 (Ethernet)
RX packets 23470 bytes 29523240 (28.1 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 14767 bytes 3701181 (3.5 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

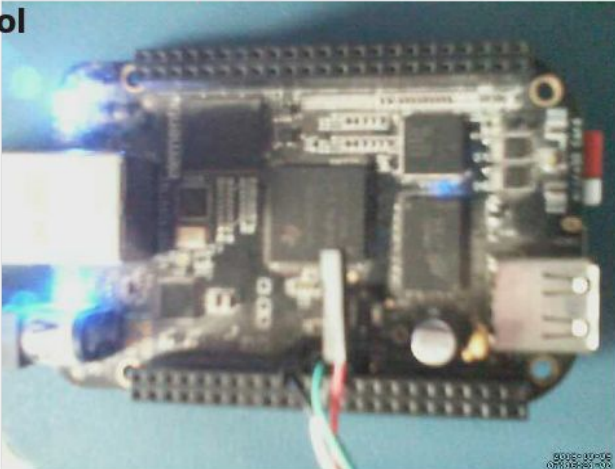
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 6006 bytes 2996787 (2.8 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 6006 bytes 2996787 (2.8 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~$

```

Don't see Streaming, wait for 10 seconds. In case streaming still doesn't start, click here.
After clicking this link Streaming should start within 7-10 seconds.

Capture Screenshot



Power Control

Live Streaming

Audio Source: MIC1-Microsoft LifeCam HD-3000

```

Hit any key to stop autoboot: 0
U-Boot> run bootcmd
gmac0: PHY present at 0
gmac0: link up, 100Mbps full-duplex (lpa: 0xc5e1)
Using gmac0 device
TFTP from server 192.168.111.1; our IP address is 192.168.111.3
Filename 'upload/DUT3/zImage'.
Load address: 0x22000000
Loading:
#####
#

#####
#

#####
#

#####
#
2.7 MiB/s
done
Bytes transferred = 3553480 (3638c8 hex)
gmac0: PHY present at 0
gmac0: link up, 100Mbps full-duplex (lpa: 0xc5e1)
Using gmac0 device
TFTP from server 192.168.111.1; our IP address is 192.168.111.3
Filename 'upload/DUT3/sama.dtb'.
Load address: 0x21000000
Loading: ###
2.1 MiB/s
done
Bytes transferred = 32408 (7e98 hex)
Kernel image @ 0x22000000 [ 0x000000 - 0x3638c8 ]
## Flattened Device Tree blob at 21000000
Booting using the fdt blob at 0x21000000
Loading Device Tree to 3fb1b000, end 3fb25e97 ... OK

Starting kernel ...

Booting Linux on physical CPU 0x0
Linux version 4.1.0-linux4sam_5.3-00053-gaa504b4-dirty
(timesys@slave8) (gcc version 5.3.0 (GCC)
) #1 Wed Aug 10 12:00:36 EDT 2016
CPU: ARMv7 Processor [410fc051] revision 1 (ARMv7), cr=10c53c7d
CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing
instruction cache
Machine model: Atmel SAMA5D2 Xplained TM43xx
cma: Reserved 64 MiB at 0x38000000
Memory policy: Data cache writeback
CPU: All CPU(s) started in SVC mode.
Built 1 zonelists in zone order, mobility grouping on. Total

```

Console log

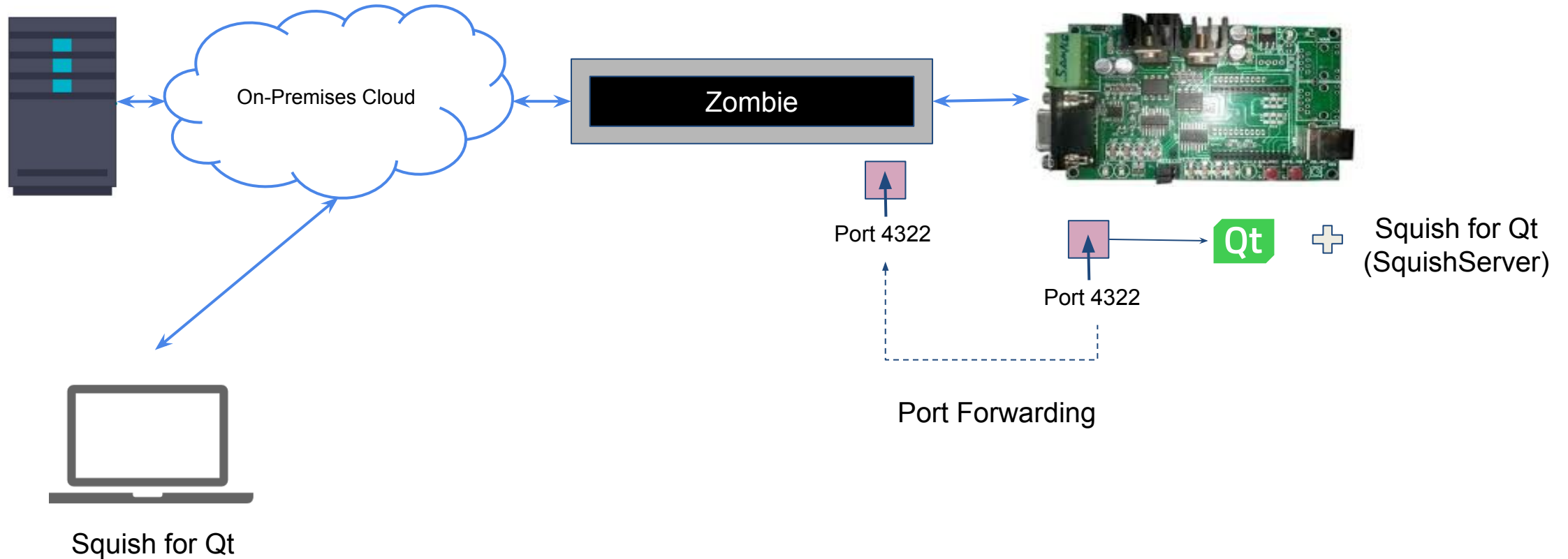
Benefits of an On-Premise Board Farm Cloud

- **Private behind your firewall** — totally under your control
- **Make board at remote locations available to everyone** — by different functional teams even with limited availability of development boards
- **Remote board debugging** — similar user experience as the local board
- **Remote viewing (Live Video/Audio streaming)**
- **Infrastructure for CI/CT** — easy to bring agile lifecycle management for your products
- **Multi-user access to shared resources** — minimize cost and helps with the schedule
- **Improved development efficiency** — even when the teams are splits between different sites, buildings/floors, labs
- **Centralized board management**
- **Easier inventory management and tracking including health of boards**
- **IT deployment friendly**
 - Runs Linux (Linux server or standard Linux distribution)
 - Boards are in a private IP network — can enable any features including dhcp server, web server with no interference to corporate IT.

Qt Desktop Application GUI Testing

Squish for Qt with BFC (Board Farm Cloud)

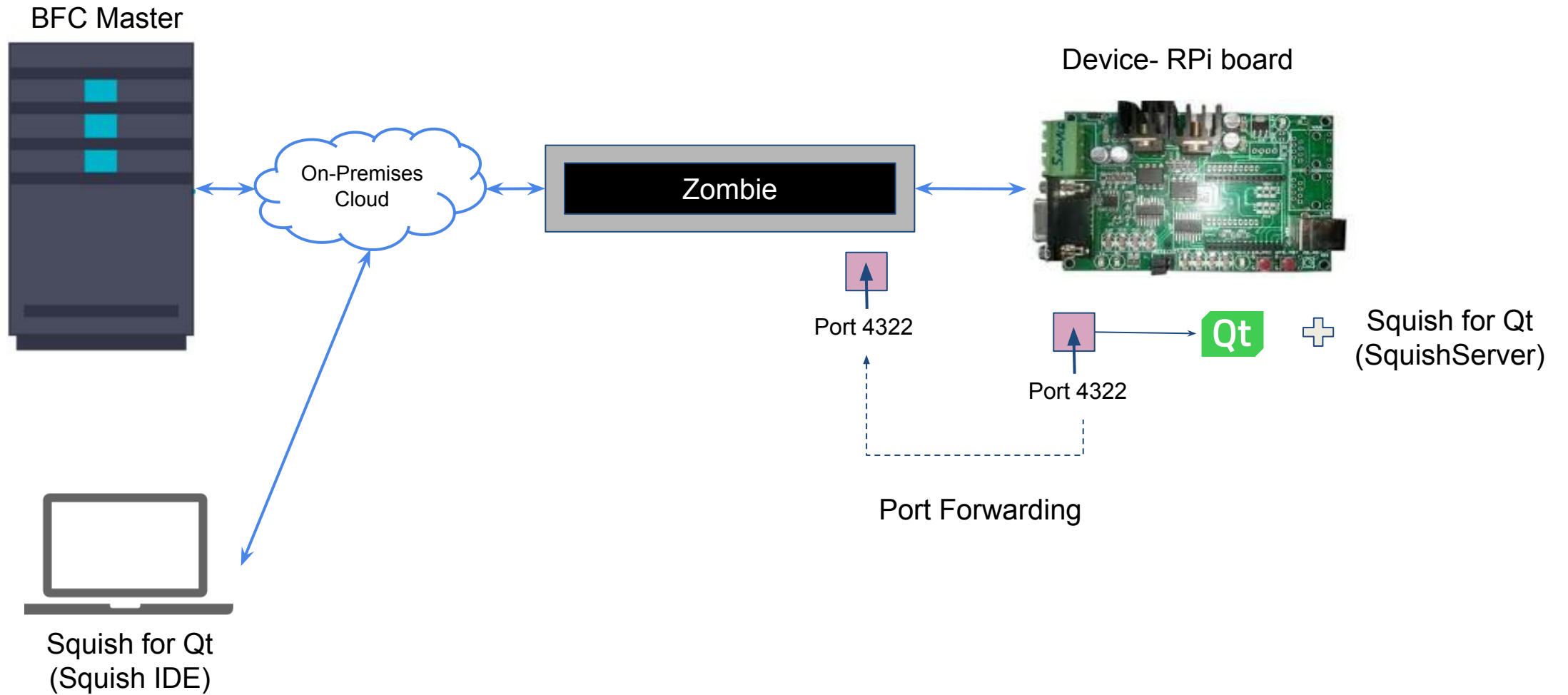
BFC Master



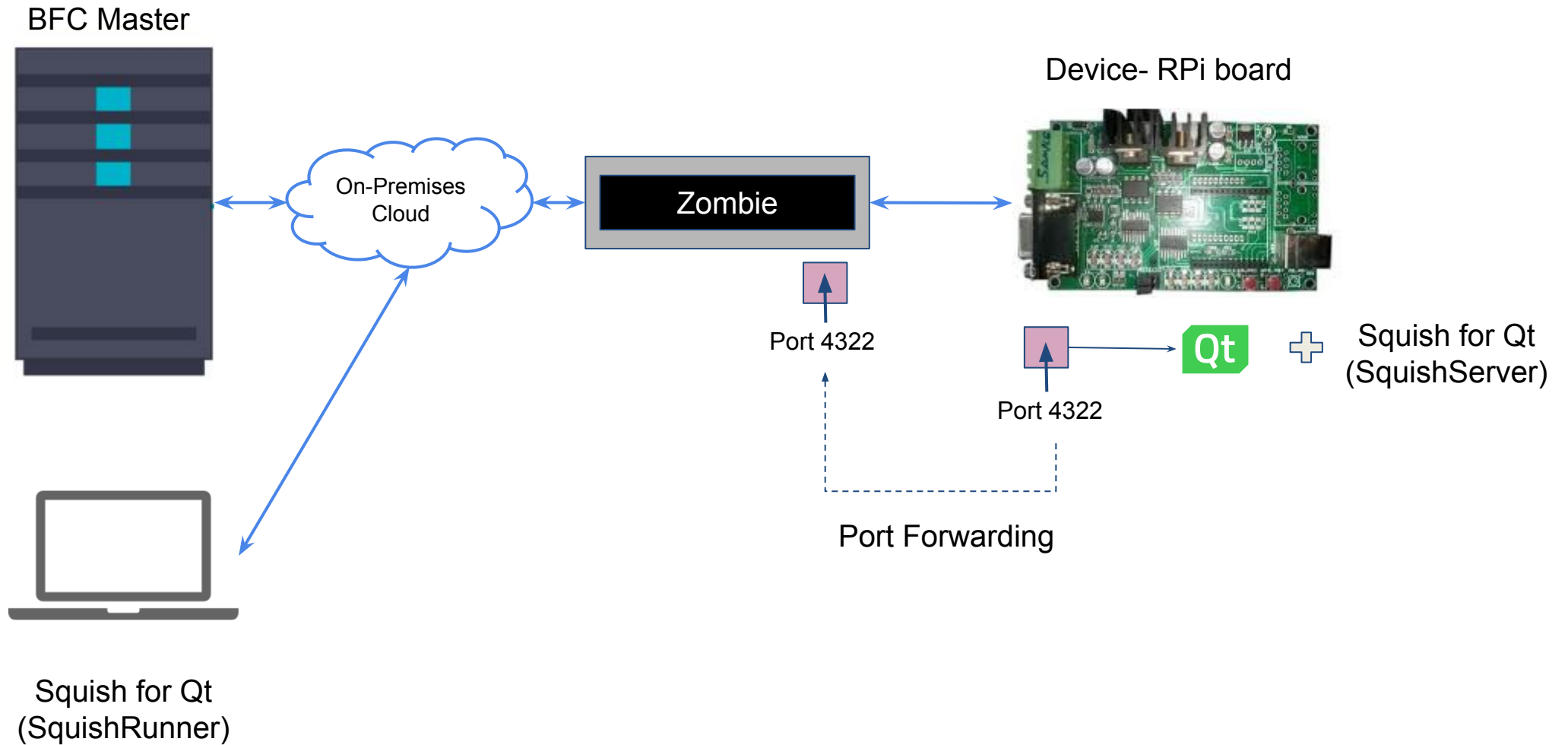
3 Modes of Test Execution

1. Squish IDE
2. CLI from remote machine
3. Continuous Testing with Jenkins

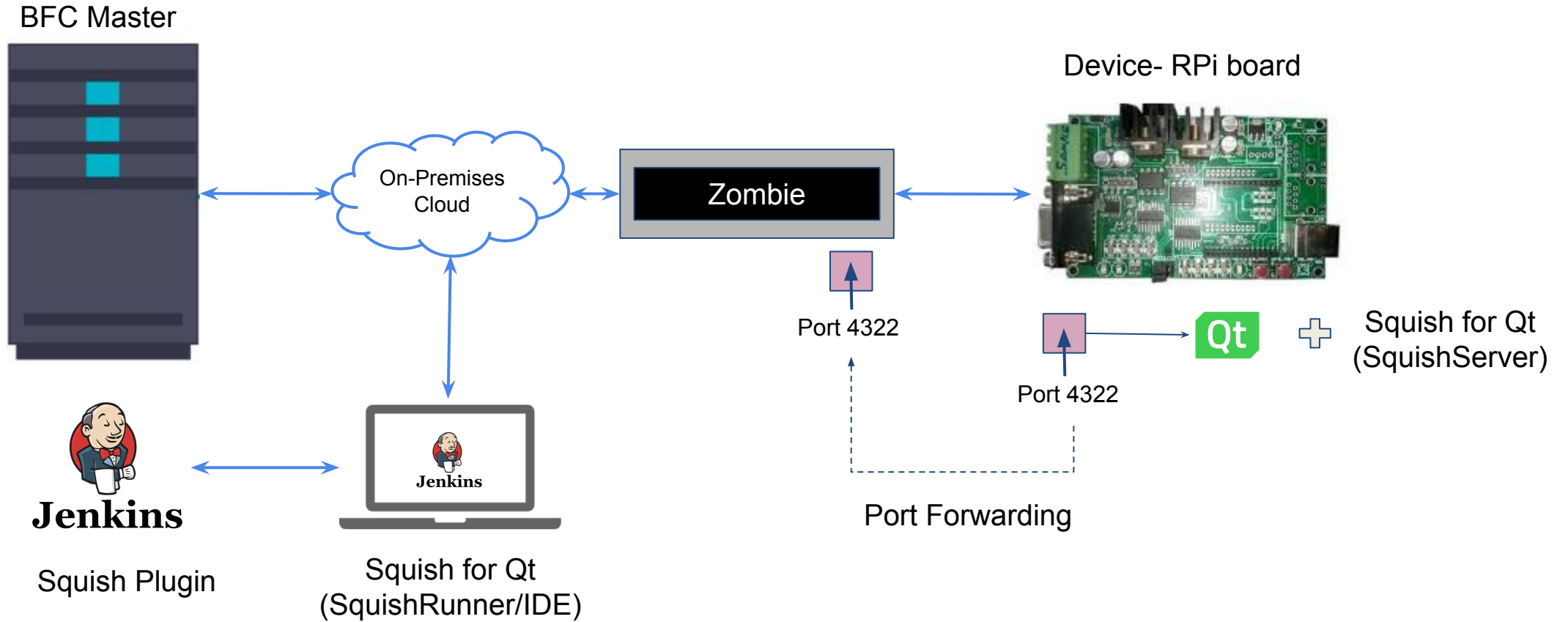
Execute GUI Test scripts using Squish IDE



Execute GUI Test scripts using CLI

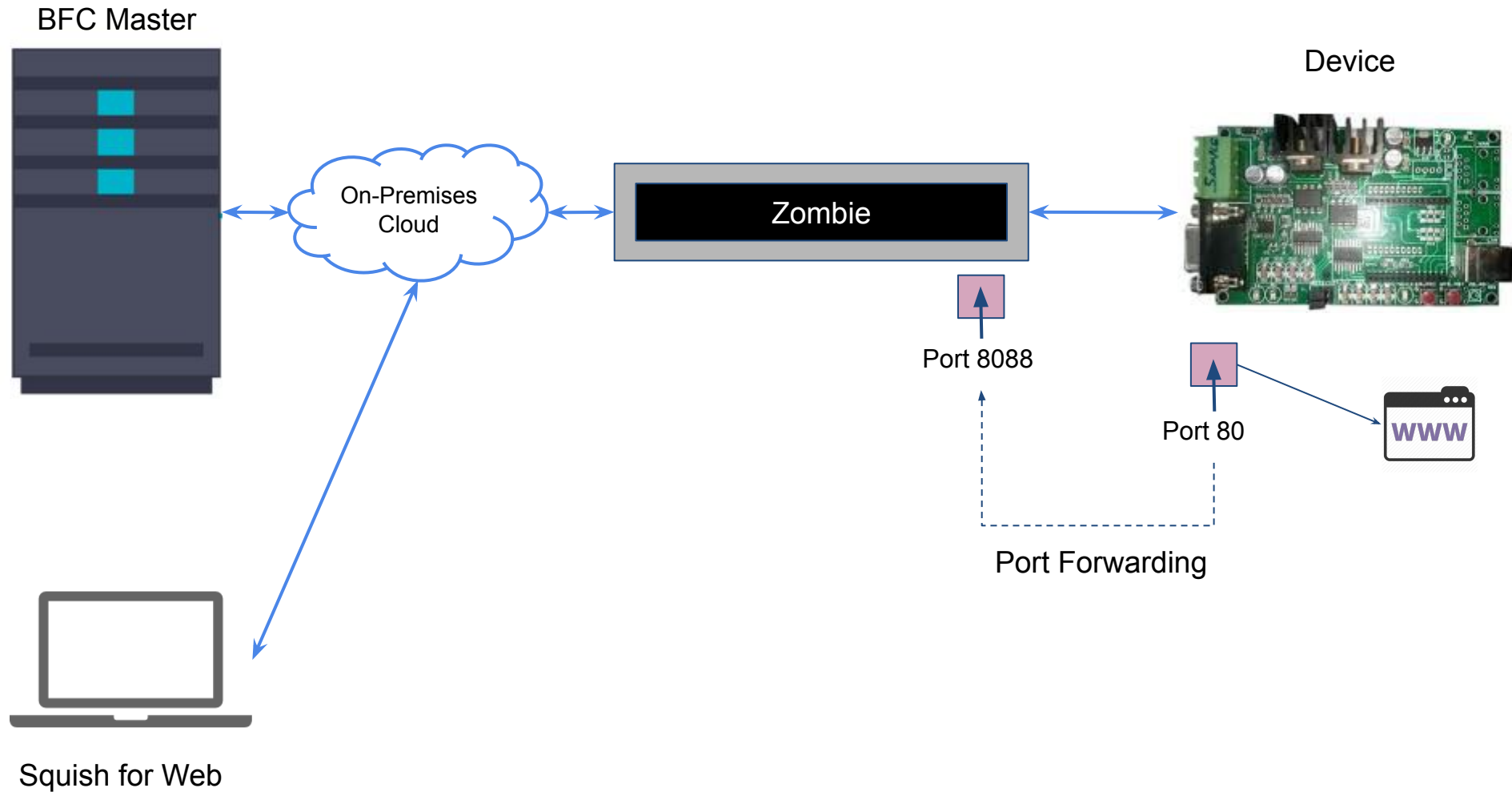


Execute GUI Test scripts using CI (Jenkins)

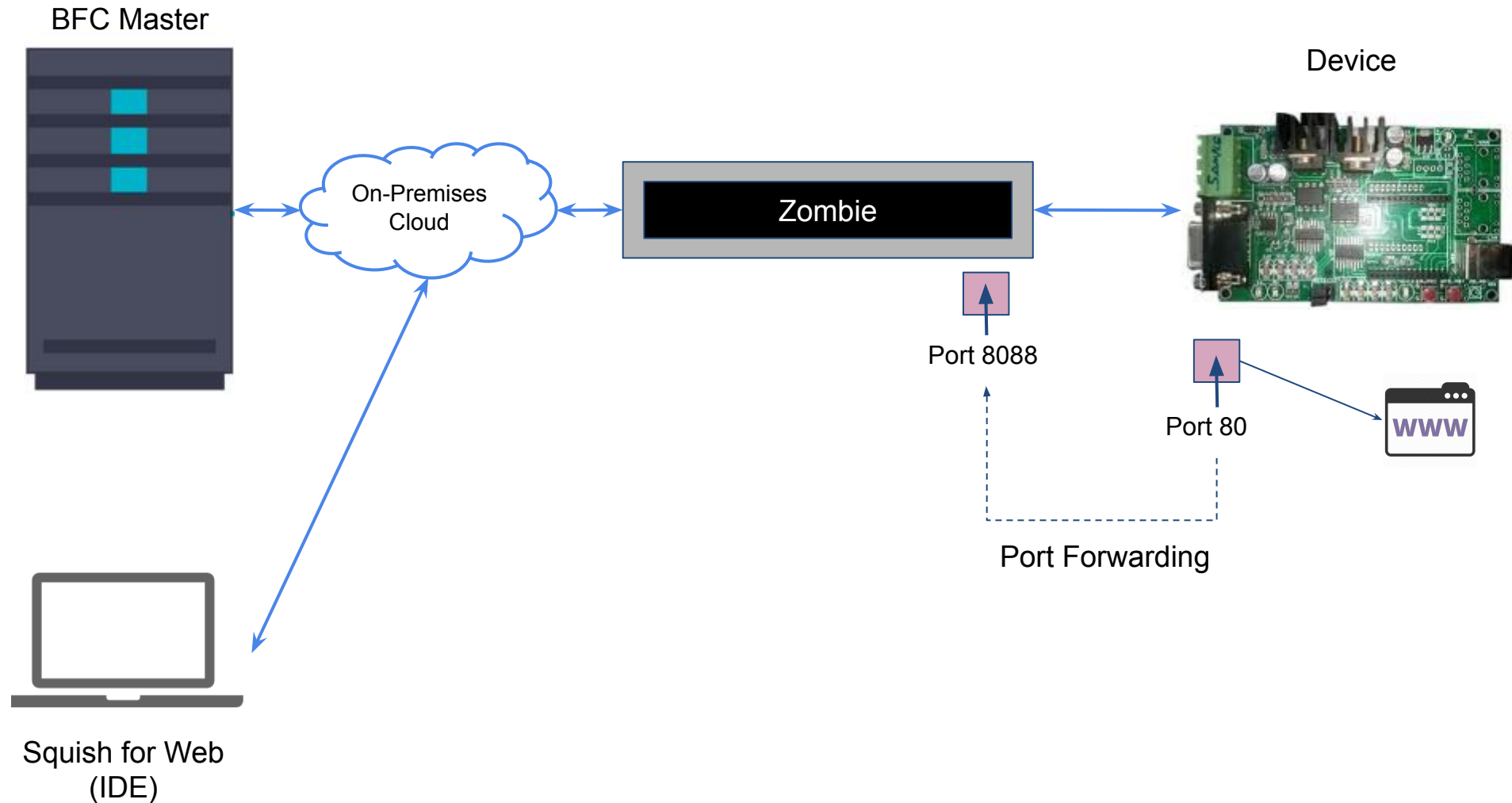


Web Application GUI Testing

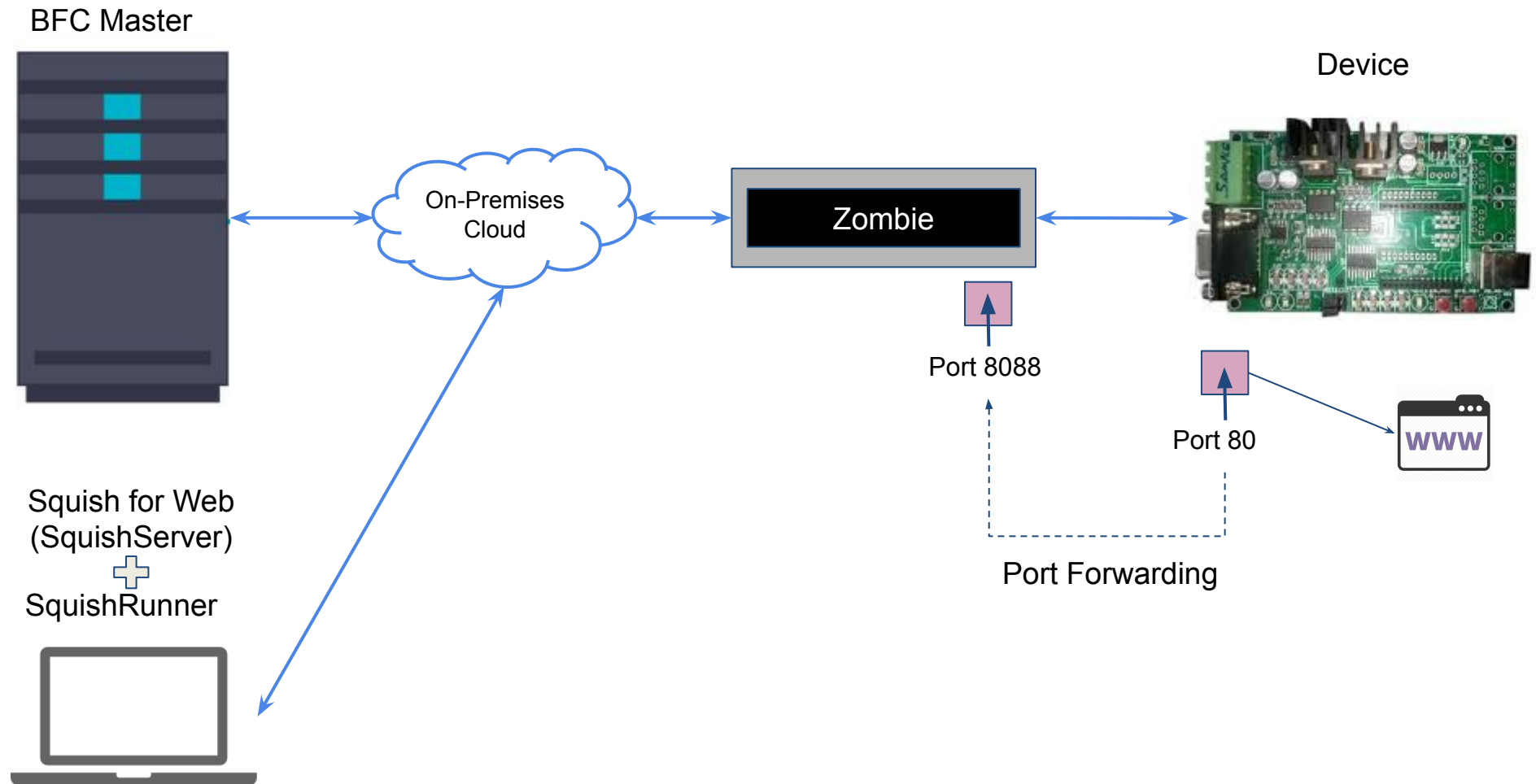
Squish for Web with BFC (Board Farm Cloud)



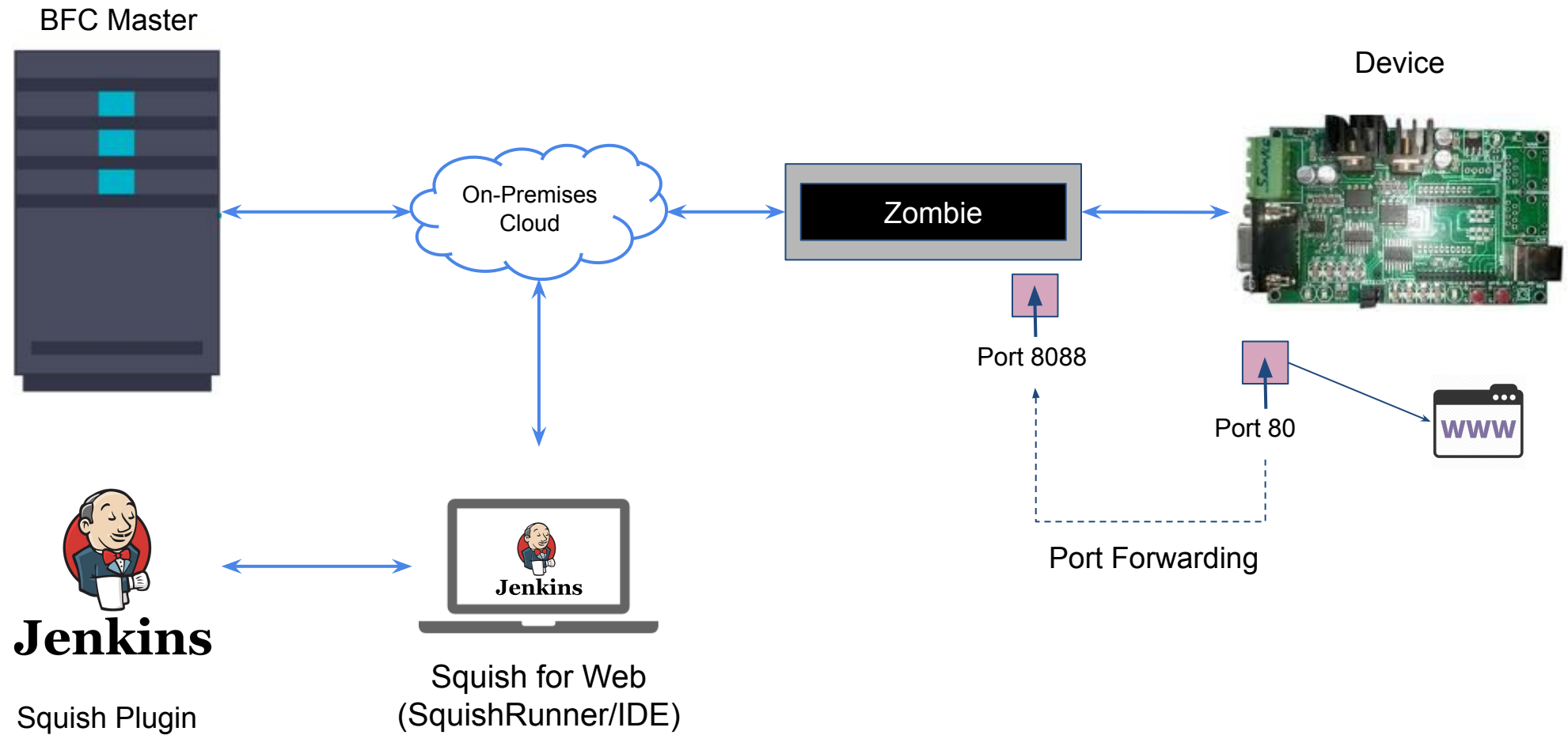
Execute Squish Test scripts using Squish IDE



Execute Squish Test scripts using CLI

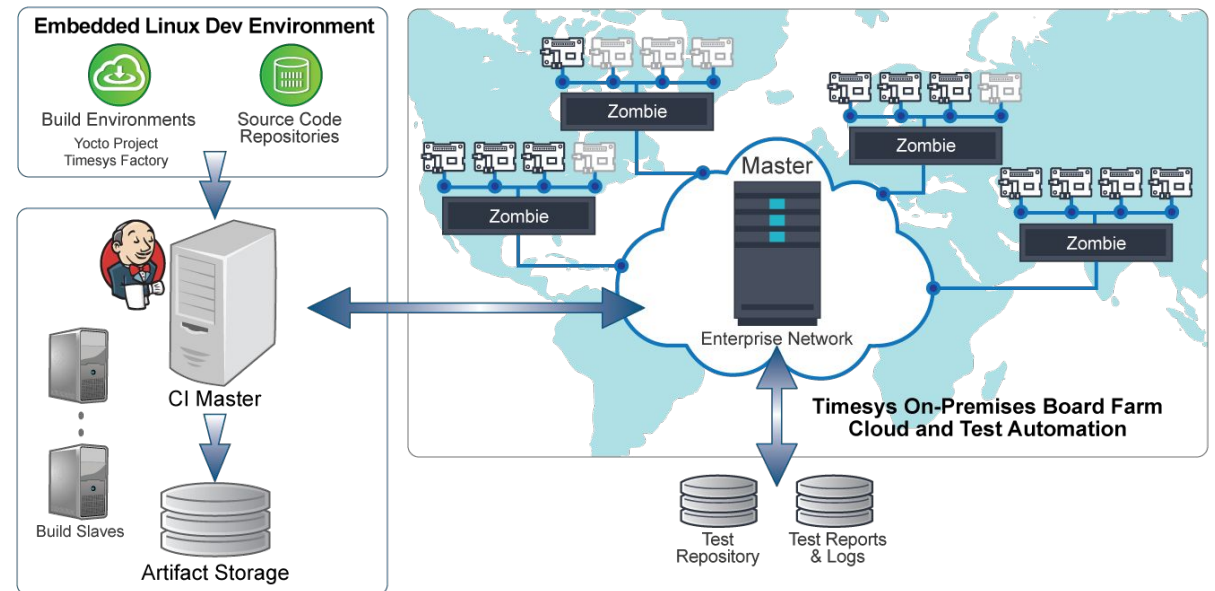


Execute Squish Test scripts using CI (Jenkins)



Timesys Test Automation Solution (TAS) with Continuous Testing

- Continuous Testing is a process of executing automated tests as part of the software delivery pipeline and feedback on the risks associated with a software release candidate as rapidly as possible.
- TAS uses:
 - Pre-integrated with CI (like Jenkins) with policy configuration options
 - Build and source code management (SCM) system independent
 - Automated test framework
 - Manual tests automated
 - Test both applications and the embedded pieces separately
 - Stress and performance benchmarking of the complete system
 - Auto report generation



Q & A

Thank You