Synchronization in an NFV World

Calnex

www.calnexsol.com

Synchronization in an NFV World

Calnex

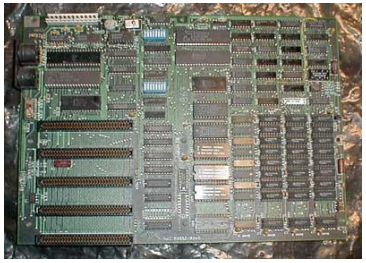
- What are Virtualisation and Network Function Virtualisation?
- Standards for NFV
- Why does NFV affect synchronization?
- Challenges and Questions
- Summary

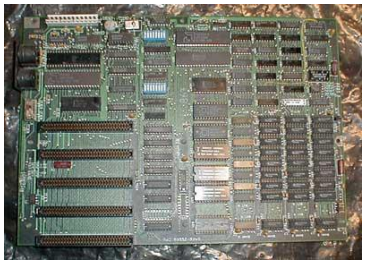
Calnex

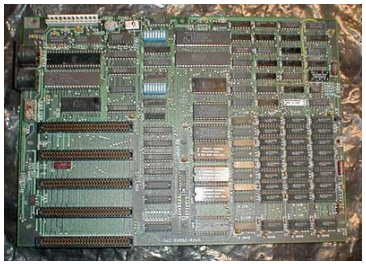
What are Virtualisation and Network Function Virtualisation?

Virtualisation

Doing in software what is traditionally done in hardware by emulating the hardware

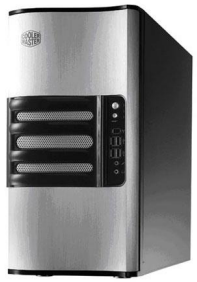






We replace this

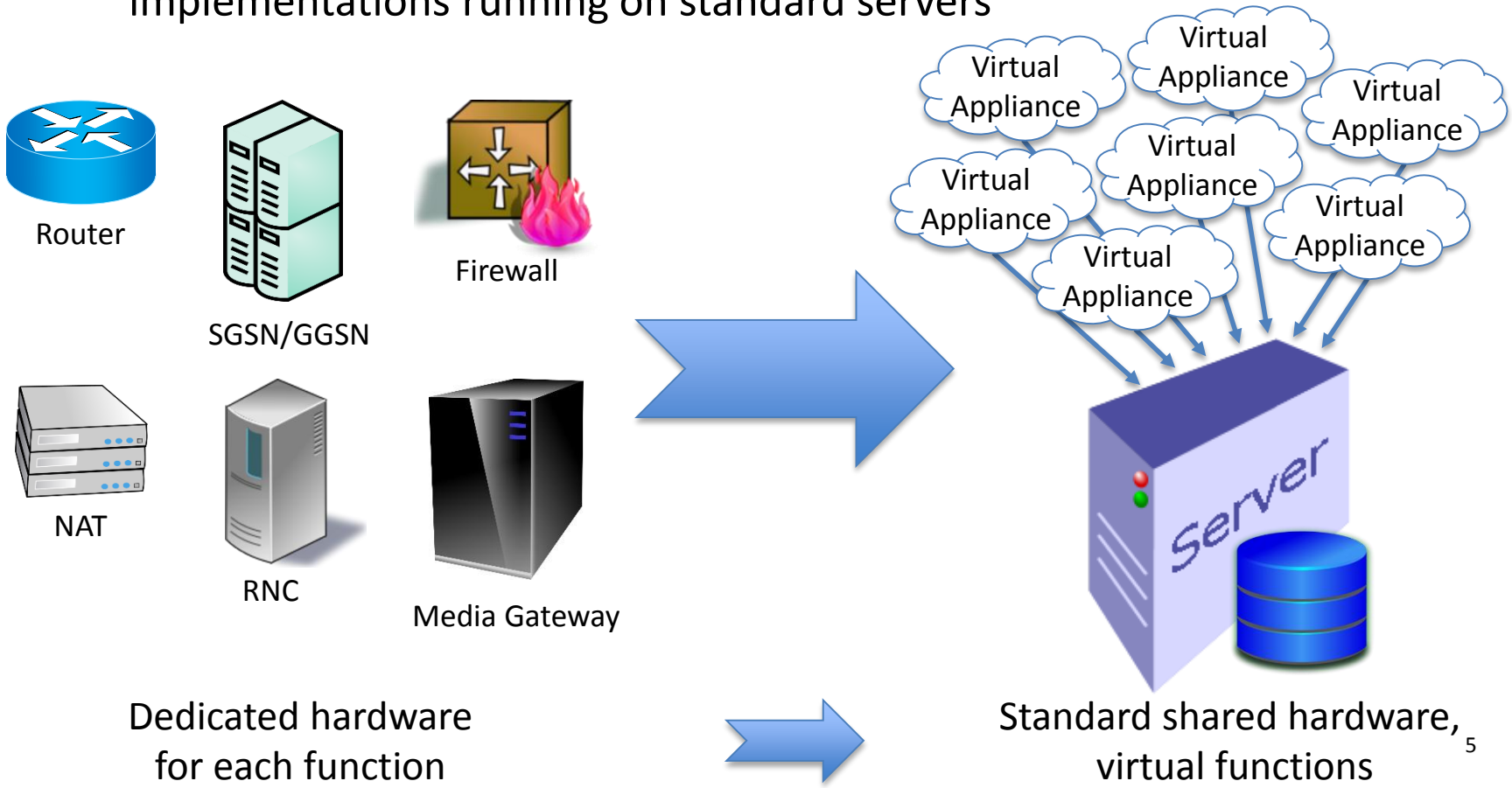
```
                'resource_id'    => $resource_details['id'],
            );
if ( $this->rule_exists( $resource_details['id'], $role_details[
    if ( $access == false ) {
        // Remove the rule as there is currently no need for it
        $details['access'] = !$access;
        $this->_sql->delete( 'acl_rules', $details );
    } else {
        // Update the rule with the new access value
        $this->_sql->update( 'acl_rules', array( 'access' =>
    }
    foreach( $this->rules as $key=>$rule ) {
        if ( $details['role_id'] == $rule['role_id'] &&
            if ( $access == false ) {
                unset( $this->rules[ $key ] );
            } else {
                $this->rules[ $key ]['access'] = $access;
```



With this

What is NFV?

- Network Functions Virtualization
  - The replacement of dedicated network elements with software implementations running on standard servers



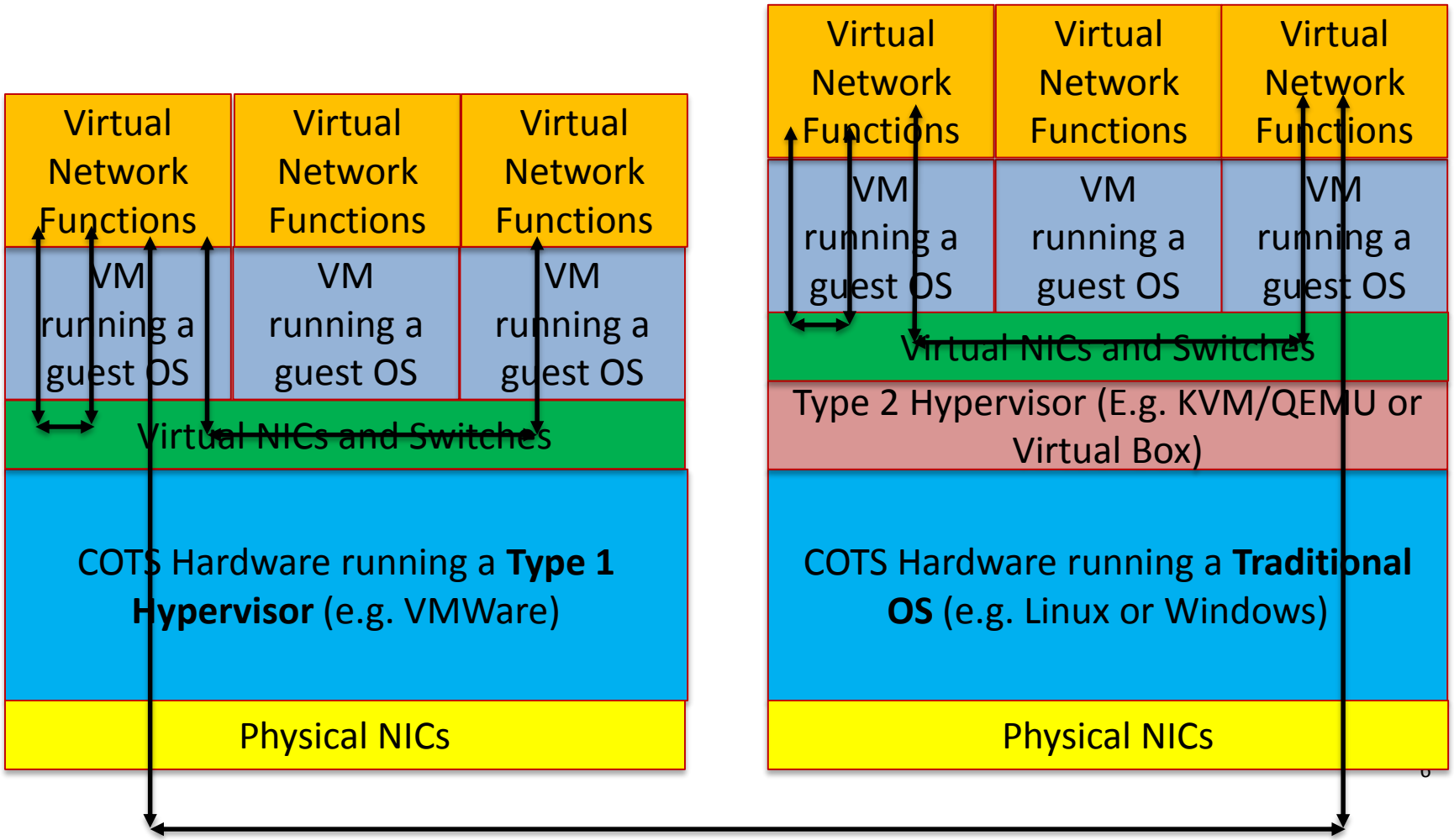
Virtualisation Layers

COTS = Commercial Off the Shelf System

OS = Operating System

NIC = Network Interface Card

VM = Virtual Machine



Why adopt NFV?

Calnex

+

Massively Increased Flexibility
Greatly Increased Speed of Deployment and Reconfiguration

Standards

ETSI NFV Reference Architecture

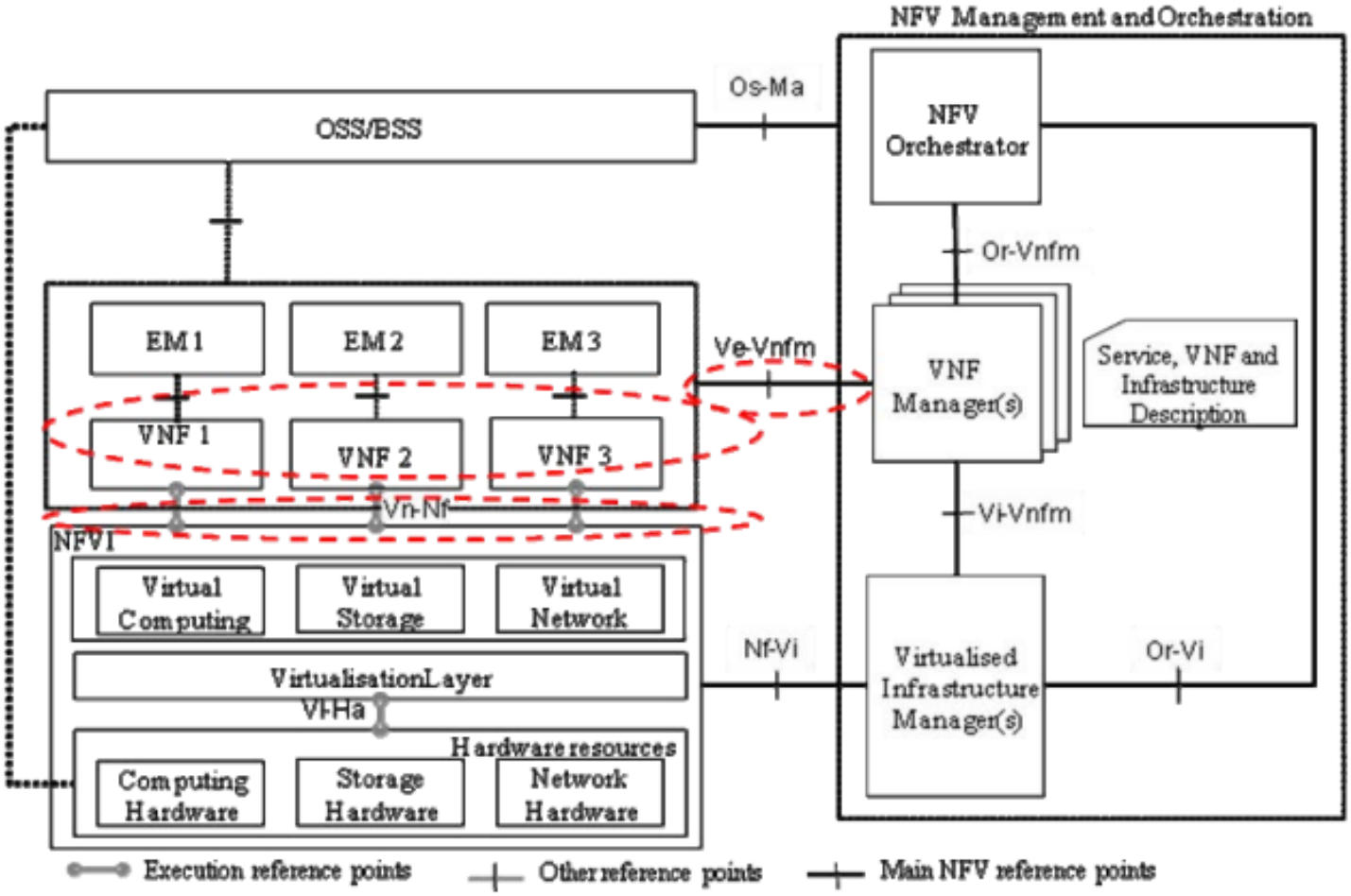


Figure 1: NFV Software Architecture Scope within the NFV Reference Architecture Framework

Standards for NFV Synchronization

Calnex

- ETSI have finalized several Standards, Recommendations and Use Cases for NFV.
  - http://www.etsi.org/technologies-clusters/technologies/nfv
- Virtualization Requirements document, Section 5.8:
  - http://www.etsi.org/deliver/etsi_gs/NFV/001_099/004/01.01.01_60/gs_NFV004v010101p.pdf
  - Service Assurance suggests the use of IEEE 1588 timestamps
  - Implemented on the NIC to establish a common time base for physical layer and upper layer processes
  - Timestamps to be used as precise time labels for all event processes

Why does it affect Synchronization?

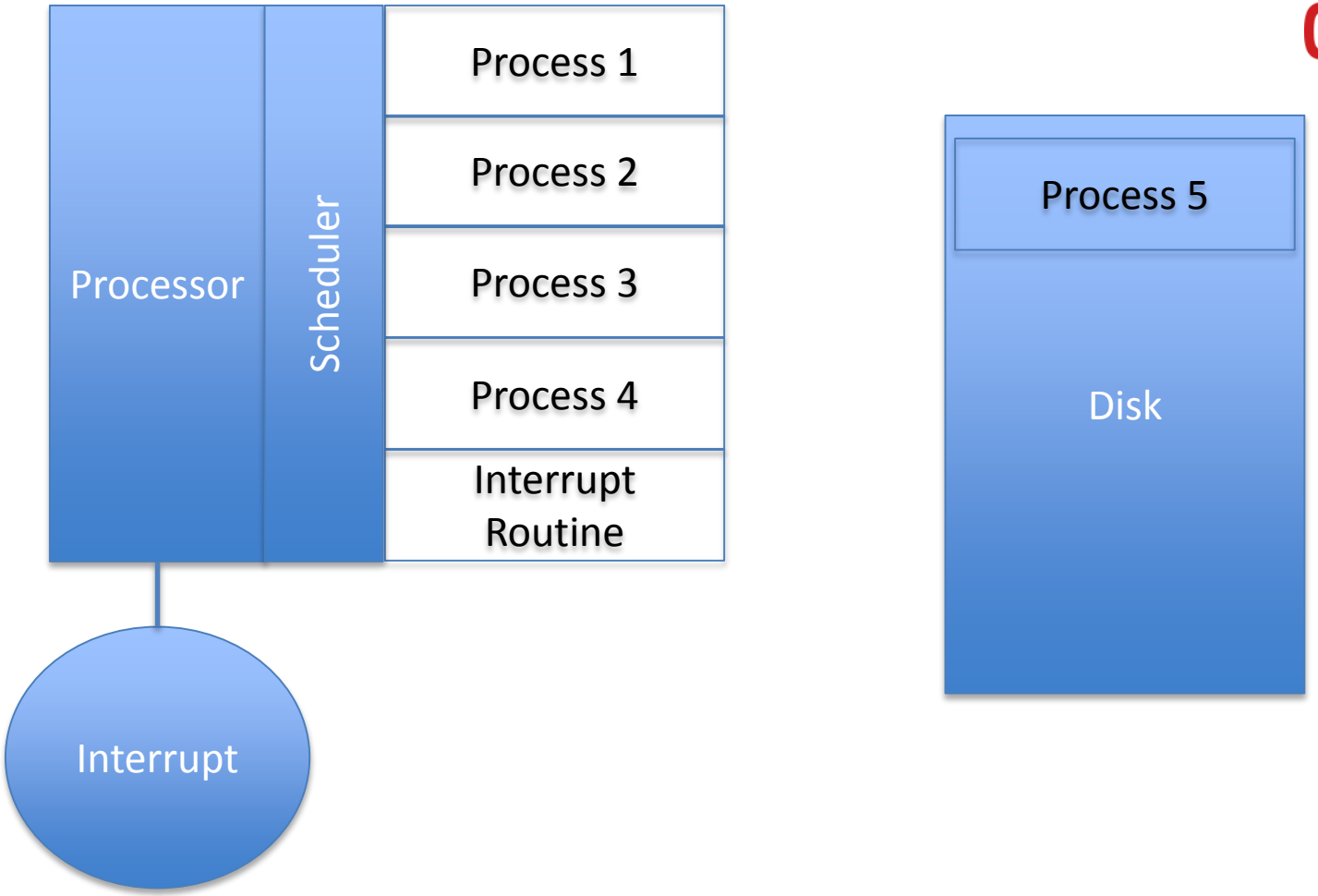
Calnex

It’s all about Accuracy

Software is both slower and much less deterministic than hardware.

The underlying processor hardware is usually clocked by a relatively low quality oscillator.

What Makes Software Less Deterministic?



The processor is usually shared between multiple processes.
The number of processes typically varies dynamically.
Processes can be swapped out to disk to make space – this takes time.
Interrupts can happen at any time – disrupting the flow of operations

What Else Makes Software Less Deterministic?

Calnex

1) Memory access times vary depending on type (cache, static, dynamic, virtual (disk)).

2) Modern processor techniques for performance optimisation such as hyper-threading, pipelining and pre-fetching branches – makes predictability difficult.

3) Multi-threading and the use of different numbers of processor cores.

4) Processors execute at different speeds – e.g. power saving vs performance modes. Different clock speeds on different machines means a given piece of software will run at a different rate on different machines.

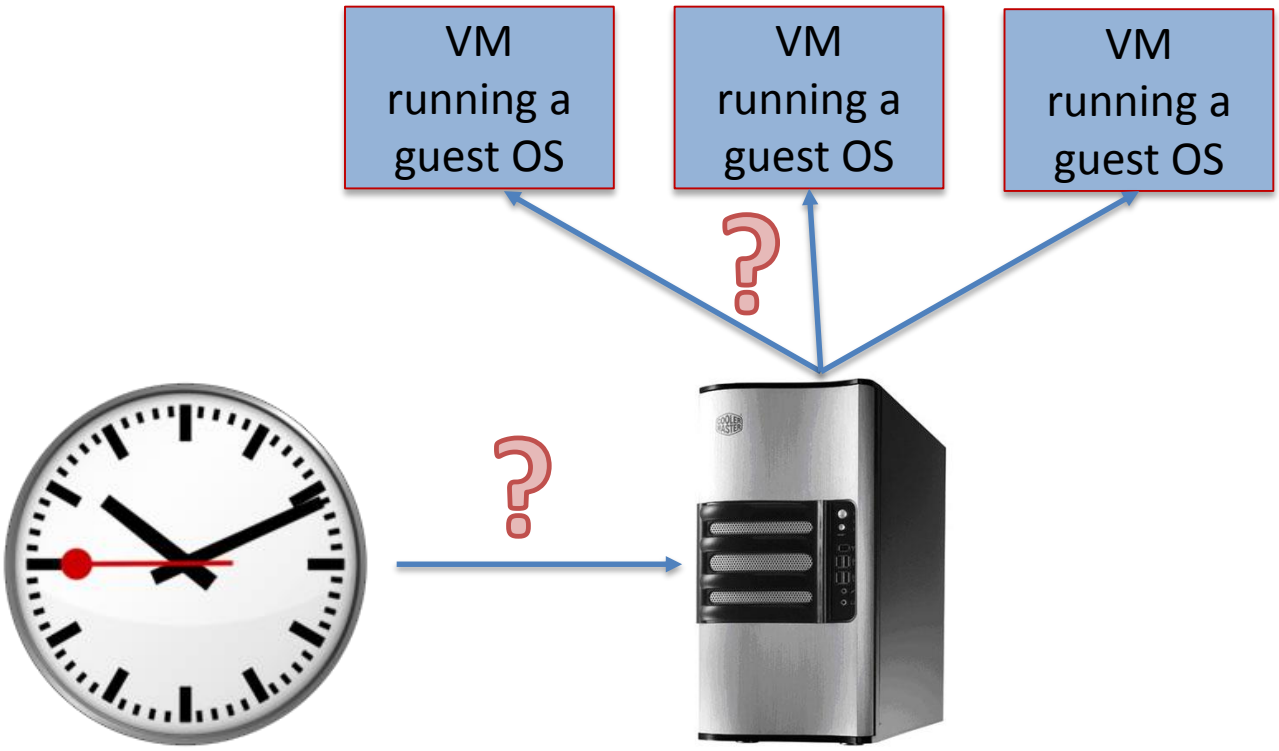
5) Software is written in high level languages which are then compiled into machine or some sort of intermediate code.

   Each time the code is changed, the sequence of lower level instructions that are executed is changed – therefore the timing changes.

Calnex

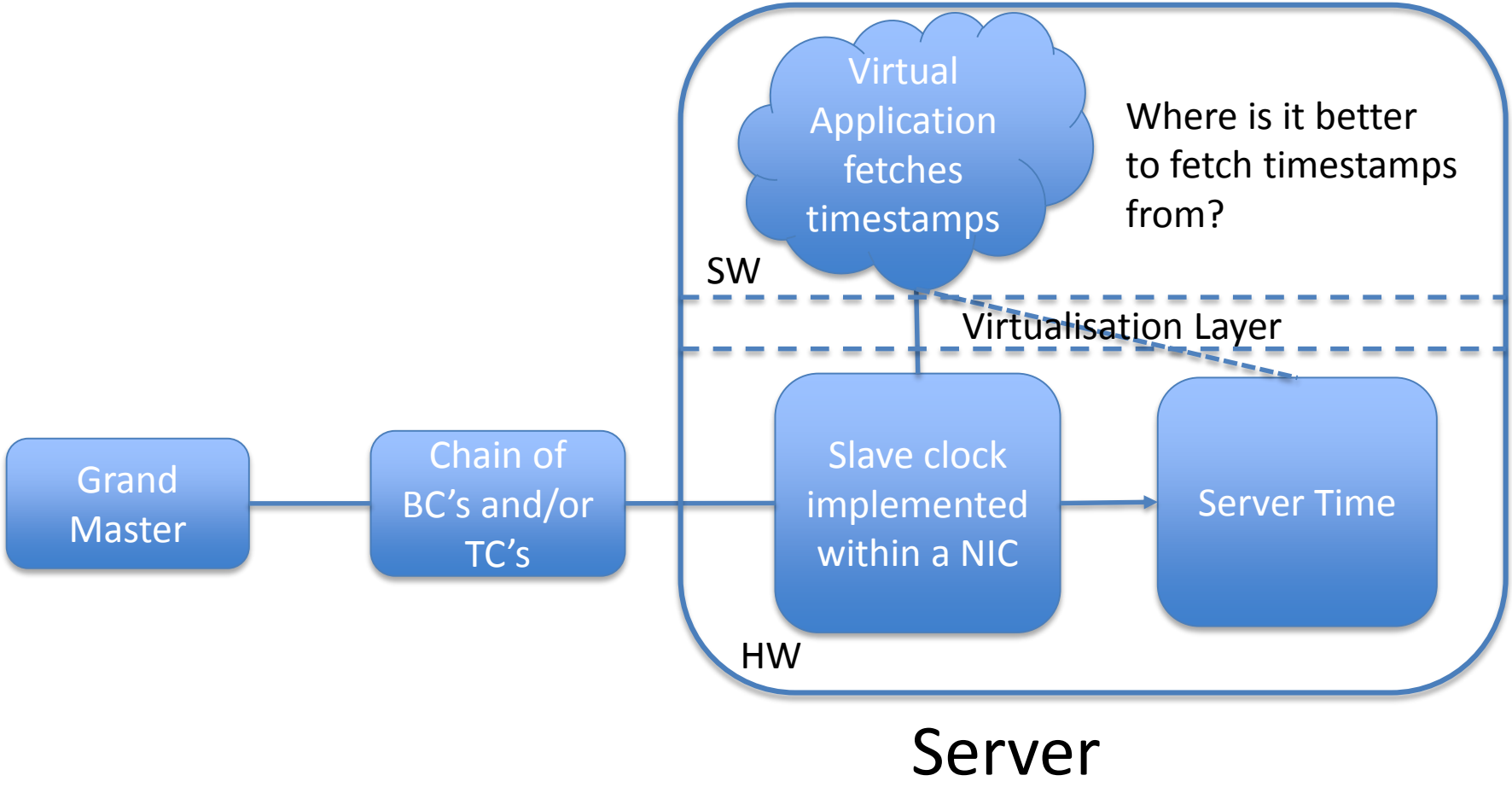
Challenges and Questions for Sync NFV

How Do We Get Suitably Accurate Time Into a VM?



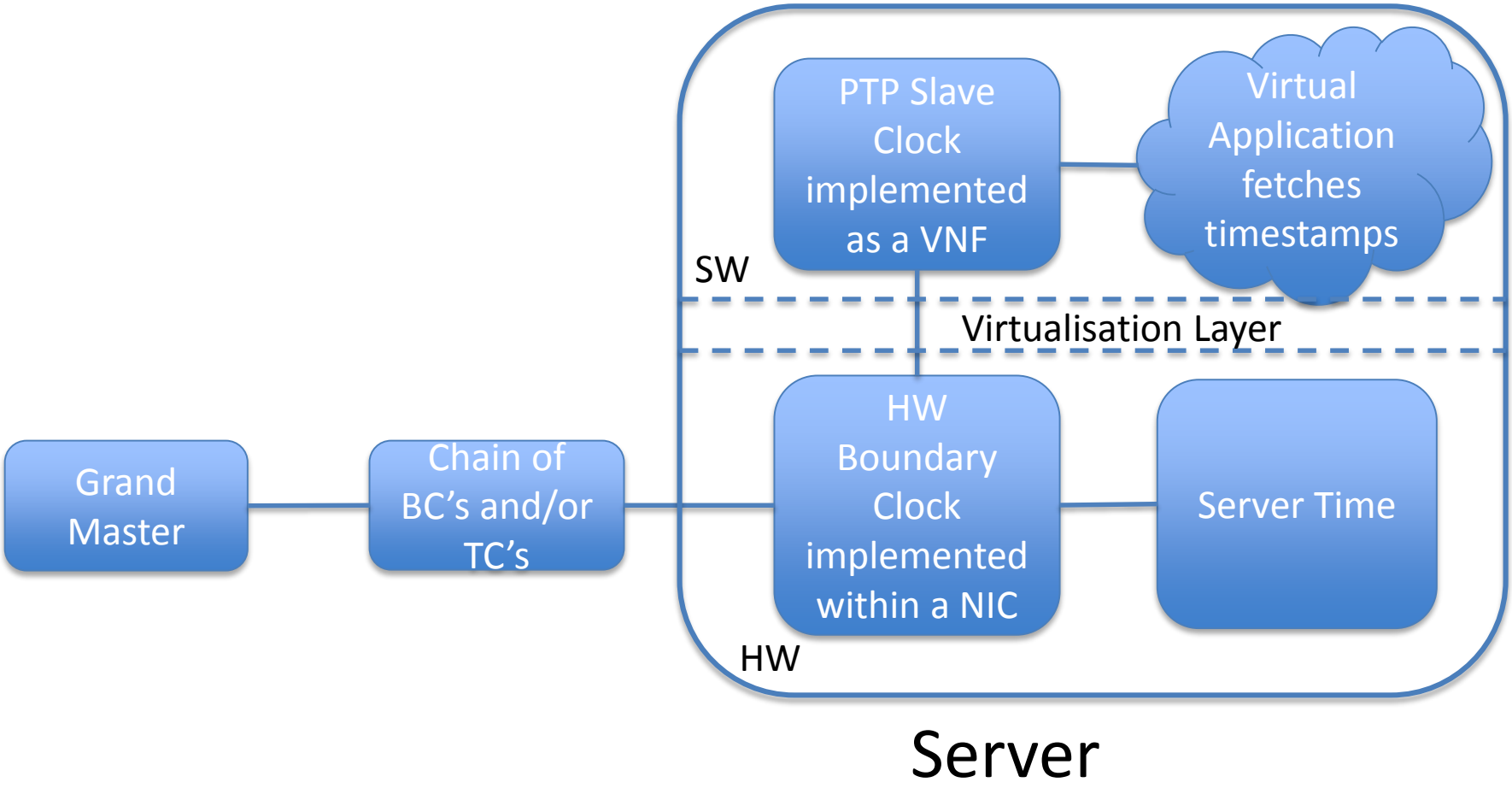
- A synchronisation chain requires dedicated hardware
- Virtualising it will not be good enough for most real world applications
- A boundary clock is a hardware function – making use of oscillators, PLL's etc.
- As soon as we cross into the software domain, things become less predictable.

A Possible PTP Synchronisation Chain



Here we have a HW synchronisation chain – the virtual function fetches timestamps from the external hardware.

A Possible Hybrid PTP Synchronisation Chain



Here we still have a HW synchronisation chain – but we have a PTP slave clock implemented as a Virtual Network Function from which the virtual function fetches timestamps as required.

Measurement is the Key

- To specify the system accuracy we have to measure it.
- The accuracy of the system is only as good as the accuracy of the measurement.
- The measurement must be traceable to an external reference.

This is a really difficult problem

Some Fundamental Problems to Solve

Calnex

1) Determination of event detection and timestamping accuracy within a VM

2) Delay characterisation and compensation within a VM

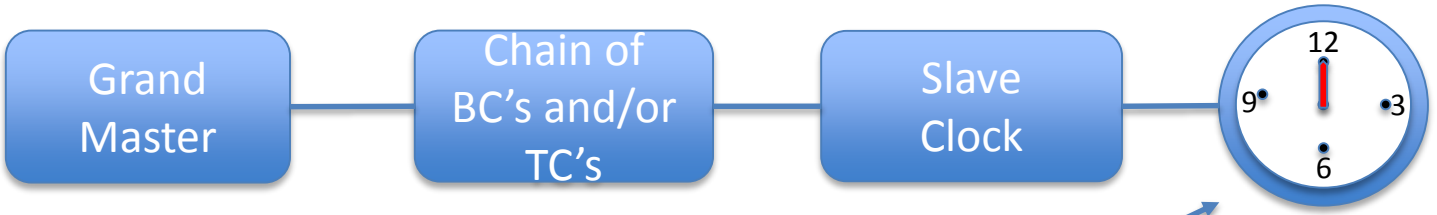
3) How do we measure timing accuracy within a VM?

Generic Event Detection and Timestamping Process

Calnex

1) Supply a timestamping engine with a suitably accurate source of time

Timestamping Engine



PTP Synchronisation Chain

11.46

3) Fetch a timestamp from the engine and store it with a record of the event.

2) Detect events that are to be timestamped.
E.g. Financial transaction of some description

Event Detection and Timestamping in Software

Loop waiting for an event

How long does it take this loop to execute? (Determines granularity)

```
While not (event_detected()){
}
```

How long after the event occurs before the actual detection? (What is the delay?)

```
Event_Time = get_timestamp();
```

When an event is detected, fetch a timestamp.
How long does this take? (What is the delay?)

The Measurement Problem

Calnex

How do we measure the accuracy of a (software) timing system inside a virtual machine?

How do we probe such a system?

How do we avoid using the system to measure itself?

How can we bring signals from inside a VM to the outside world so that we can measure them?

Measuring Software in Virtual Machines

Calnex

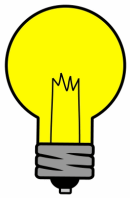
To measure software systems, they need to be instrumented – typically using software embedded within the system – therefore the instrumentation has to be included at development (compilation) time.

But we've already seen the problem of fetching accurate timestamps.

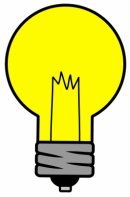
The measurement system is entangled with the system it is trying to measure – the two systems affect each other.

If we want to monitor virtual network packets, we need to provide virtual taps – but these too affect the performance – copying is a very expensive operation in terms of time – and how long it takes depends on the length of the packet.
(A lot of virtual network optimisation depends on so called "zero copy" operations.)

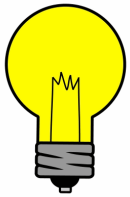
Some Ideas



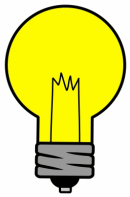
Instrument timing critical software in a standard way (e.g. define a standard API). Of course, this will affect the timing!



Define and use compiler directives to report how many machine cycles are required for timing critical operations.
Use that information at run time for determining delays.



Develop (programming and compilation) techniques for keeping time critical loops and operations as tight and predictable as possible.



Use dedicated hardware to provide/support virtual taps (e.g. duplicated memory) – at the cost of genericity.

Regardless of what we do, software will never be as accurate as hardware – but maybe for some applications it could be good enough.

Some Big Questions – Research Needed

Calnex

How accurate might we be able to make a software PTP implementation?

How accurately might we be able to transfer and maintain time within a VM with or without specialised hardware?

How can we solve the measurement problems?

Summary

Summary

Calnex

- NFV is coming, like it or not
  - Most major operators are considering it, if not actively planning for it
  - Probably the biggest shake-up of telecoms networks since voice-data convergence 10 years ago
- Synchronization will be affected
  - NFV doesn't remove the need for synchronization, but synchronization methods will need to evolve
  - New models of operation will be established
  - New opportunities will be created
  - But there are significant challenges to be overcome to
    a) Make Sync in NFV good enough for real applications
    b) Be able to measure timing inside a VM with sufficient accuracy
- There is much work to do before software can replace hardware in time critical applications (if ever.)