



# **Multi-Stream Scenarios**

## **Enhancing Stimulus Generation in the VMM**

Sumit Dhamanwala – Verilab

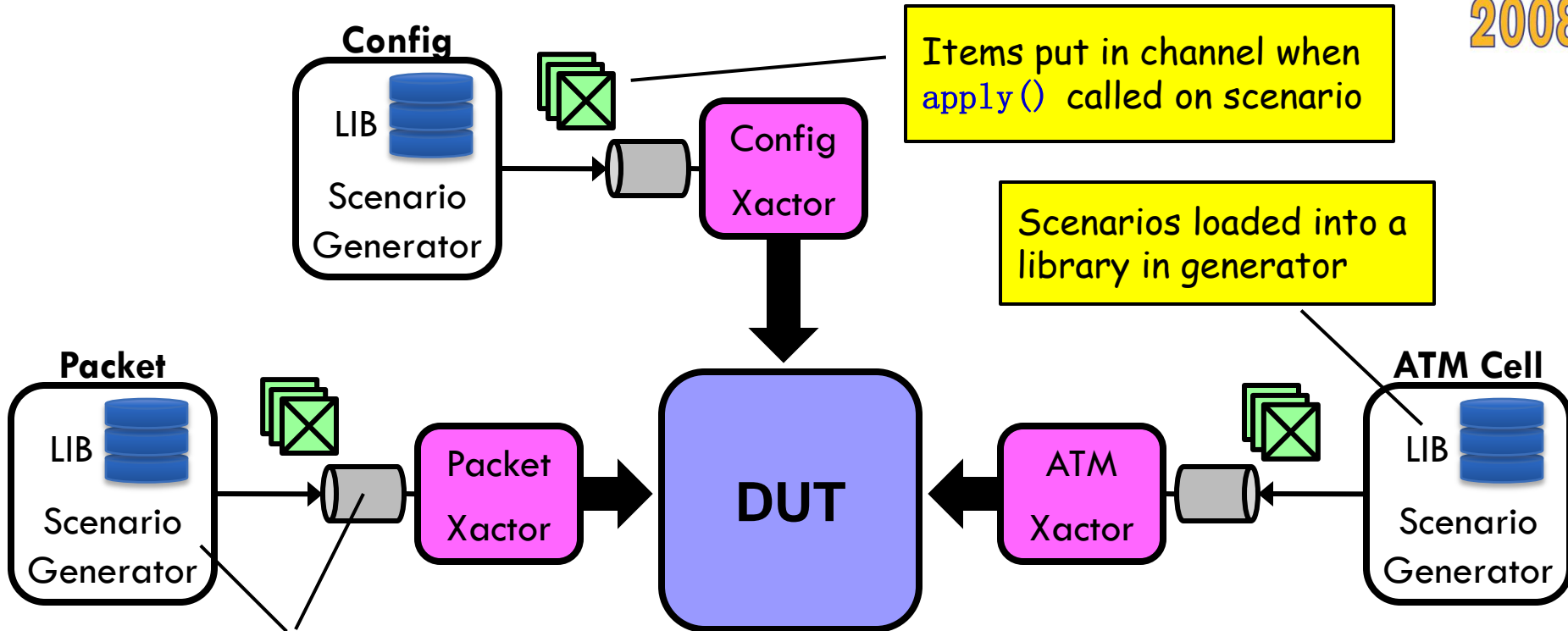
Jason Sprott – Verilab

([jason.sprott@verilab.com](mailto:jason.sprott@verilab.com))

# Today's Agenda

- Existing Solution for Stimulus Generation
- Challenges
- Channel Grab/Ungrab Additions
- Multi-stream Scenario Additions

# Scenario Generation Today

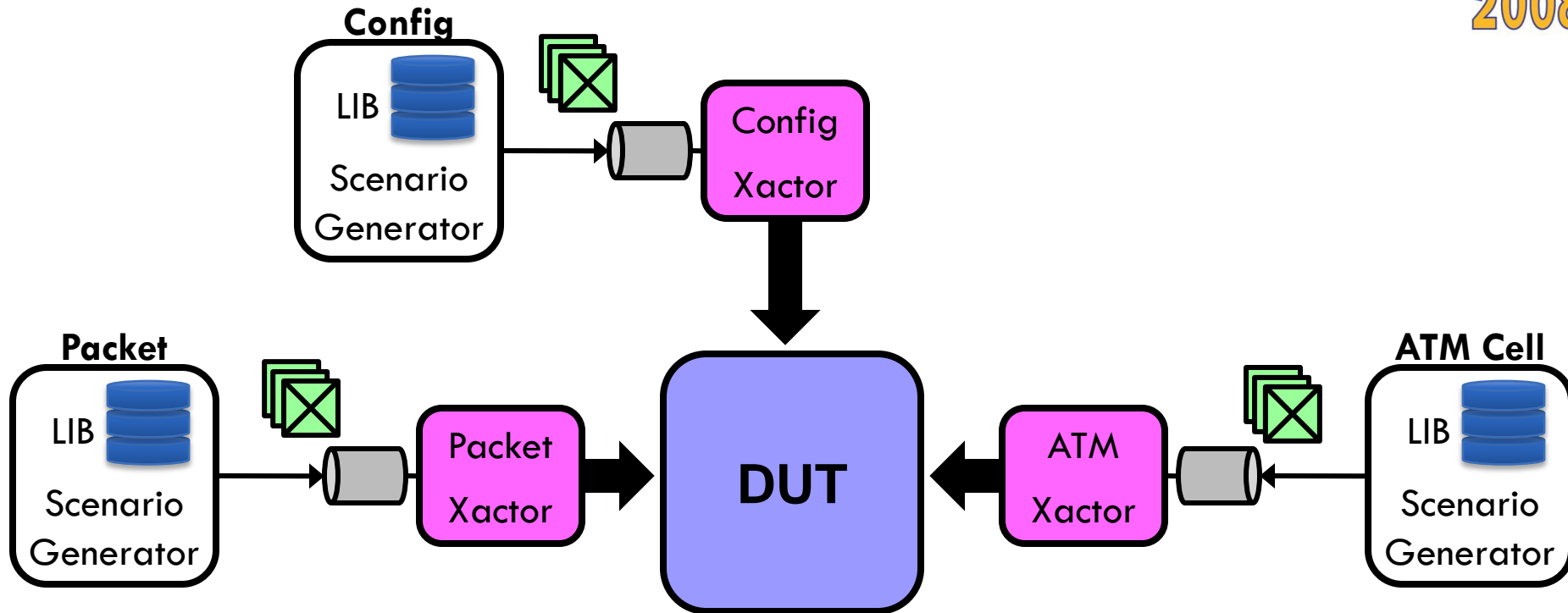


User defined channels and scenario generators:

```

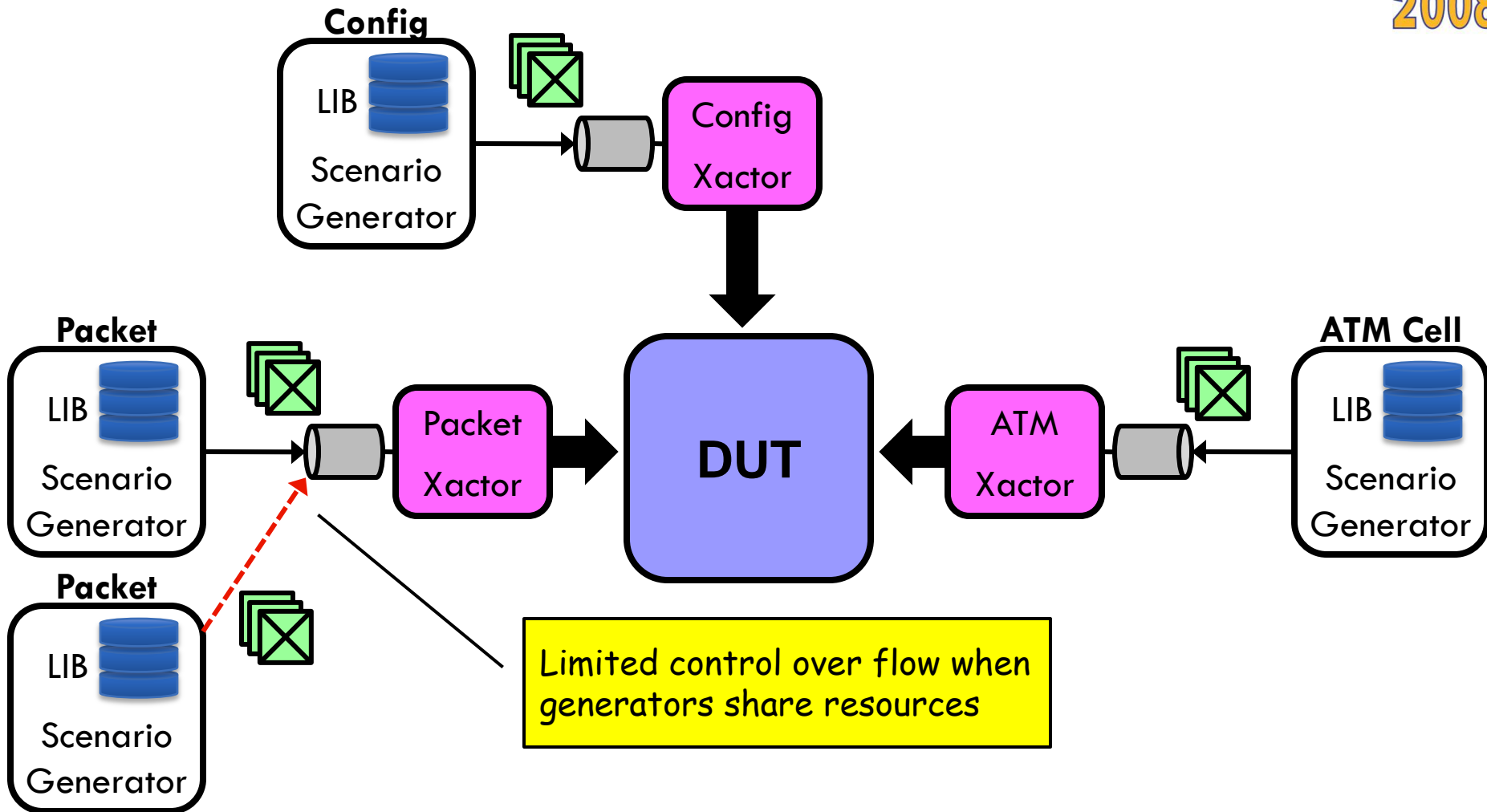
`vmm_channel(Packet)
`vmm_scenario_gen(Packet)
    
```

# Scenario Generation Today

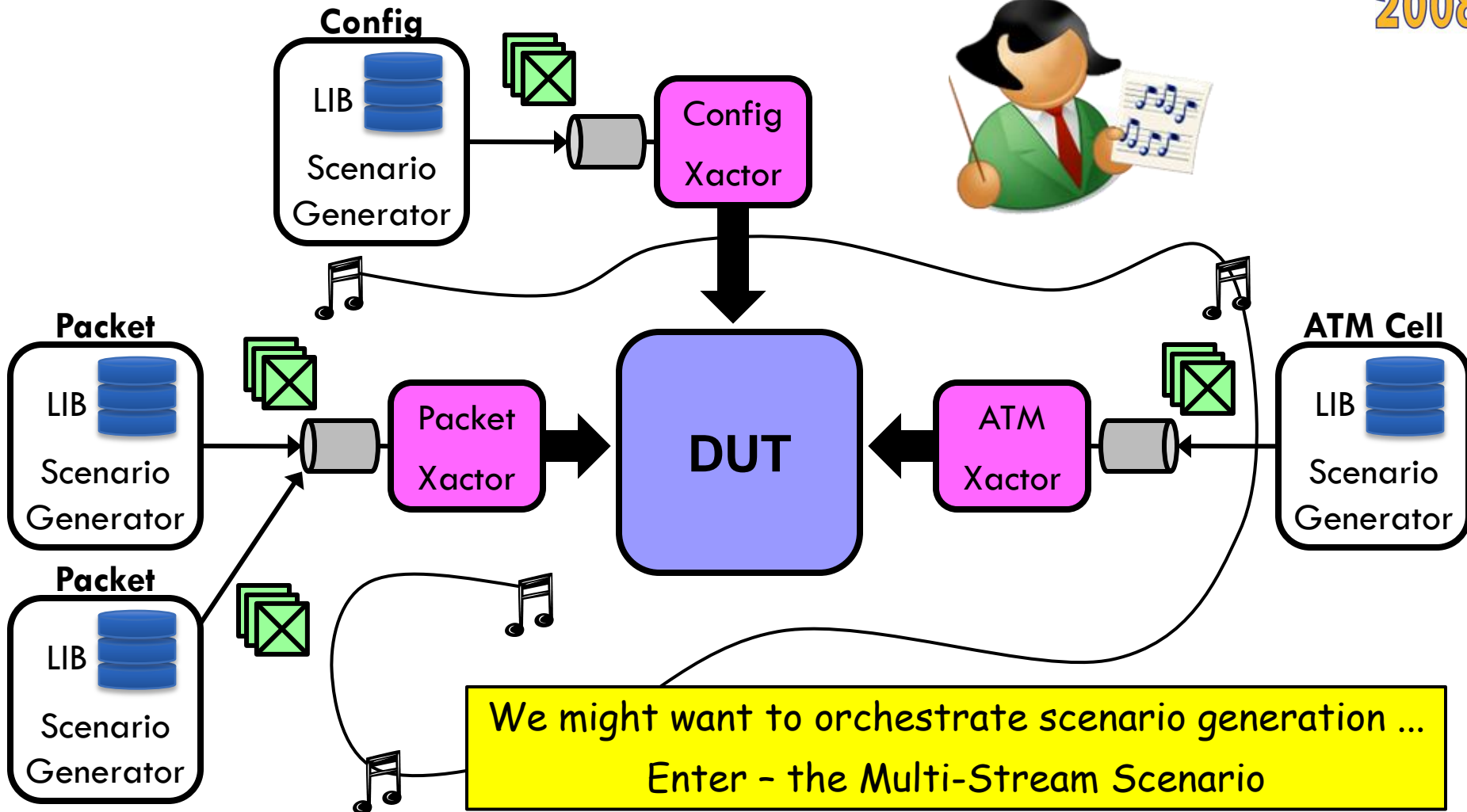


Generators all run independently

# Resource Sharing Today



# What If We Want More Control?



# The Challenges

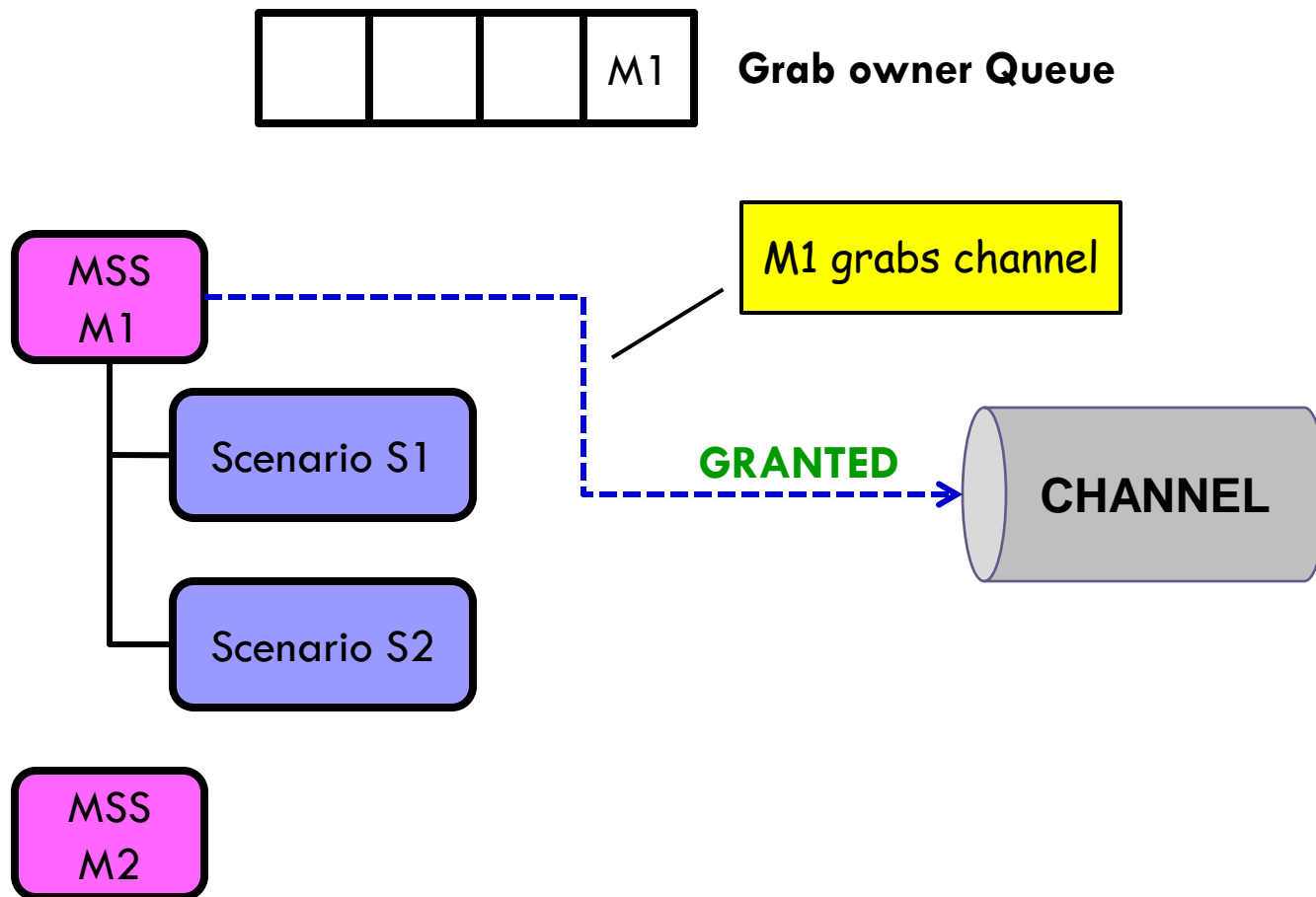
- Backward Compatibility
- Resource Sharing
  - Grabbing channels
  - Making that hierarchy aware
- Multi-stream Scenarios (MSS)
  - Composition: other MSS; scenarios; items
  - Database of accessible resources

# VMM Standard Library Modification

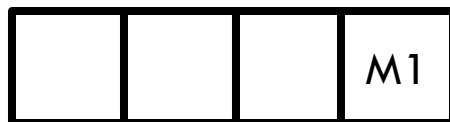
- vmm\_ms\_scenario (added)
  - Methods to establish parent/child relationship
- vmm\_ms\_scenario\_gen (added)
  - vmm\_channel registry
  - vmm\_ms\_scenario registry
  - vmm\_ms\_scenario\_gen registry
- vmm\_ms\_scenario\_election (added)
  - Selection scheme
- vmm\_scenario (modified)
  - Adding parent/child relationship for channel grab
- vmm\_channel (modified)
  - Grab/Ungrab



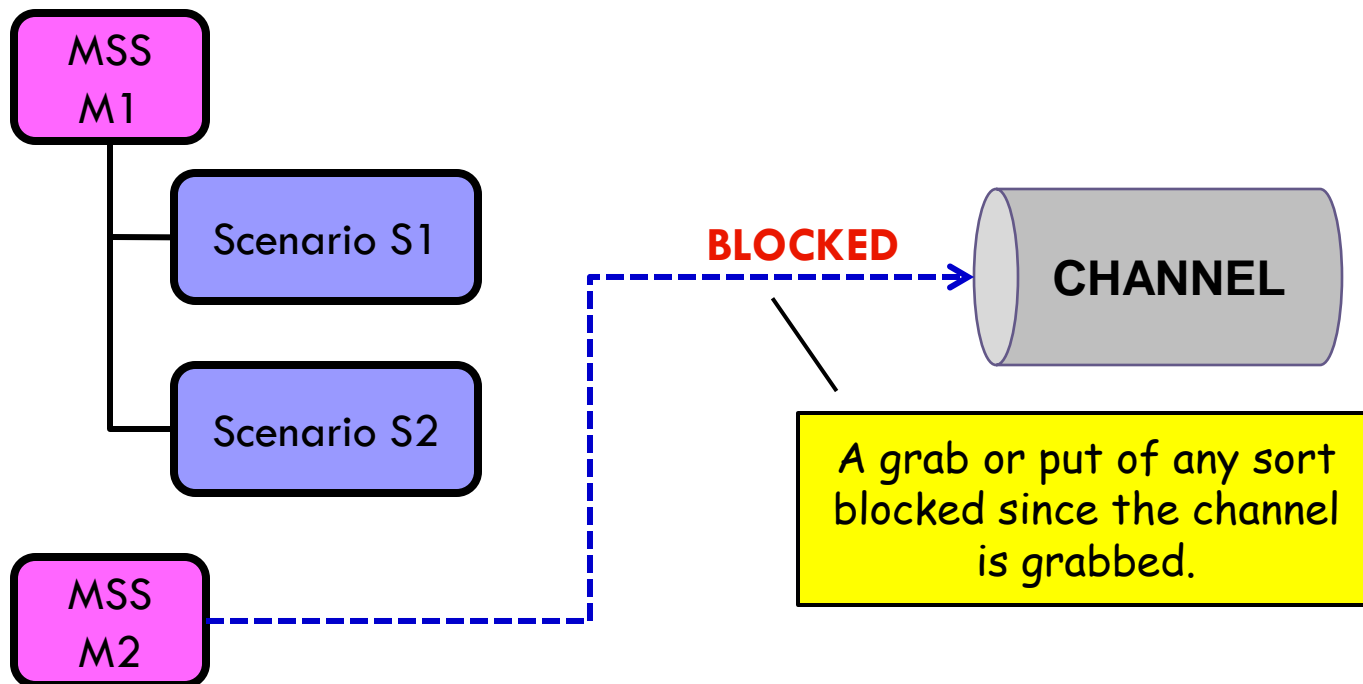
# VMM Channel Grab/Ungrab



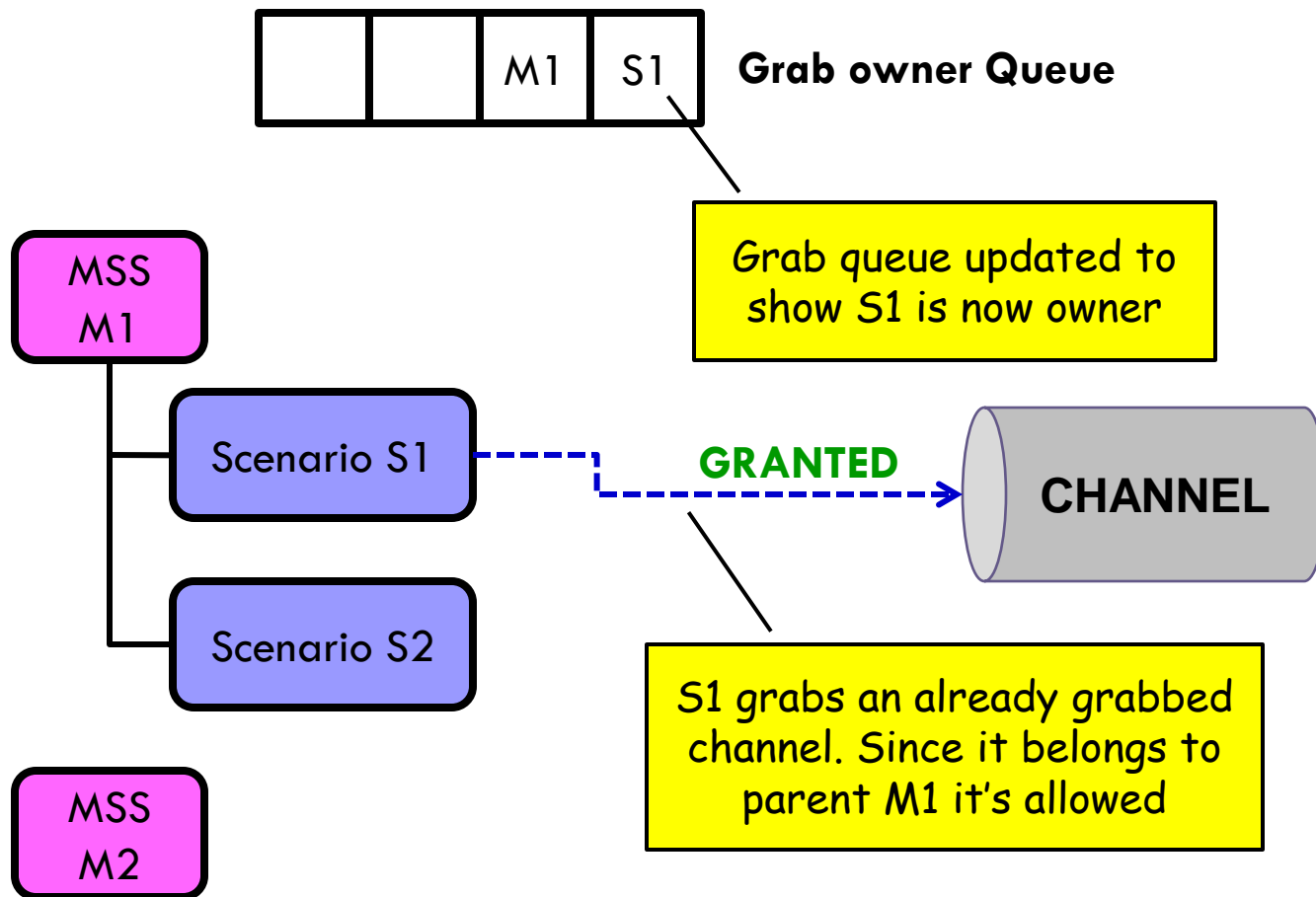
# VMM Channel Grab/Ungrab



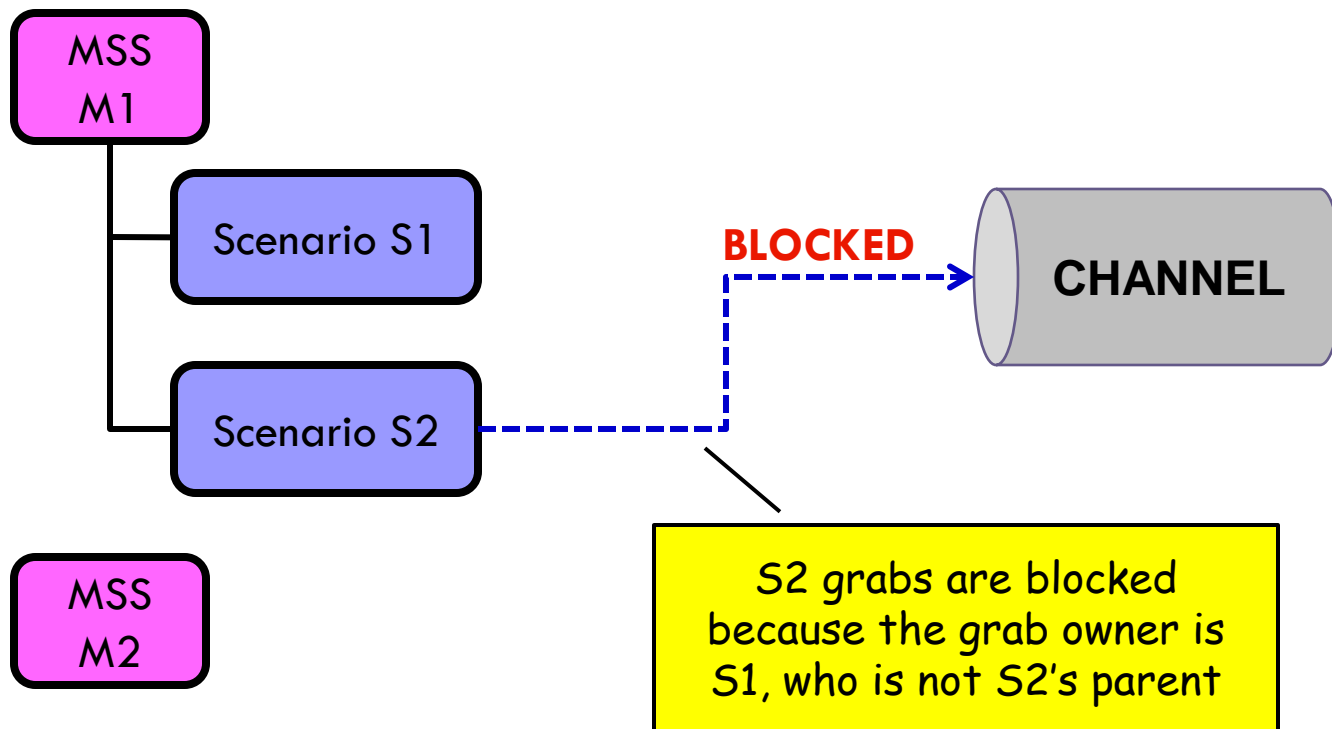
Grab owner Queue



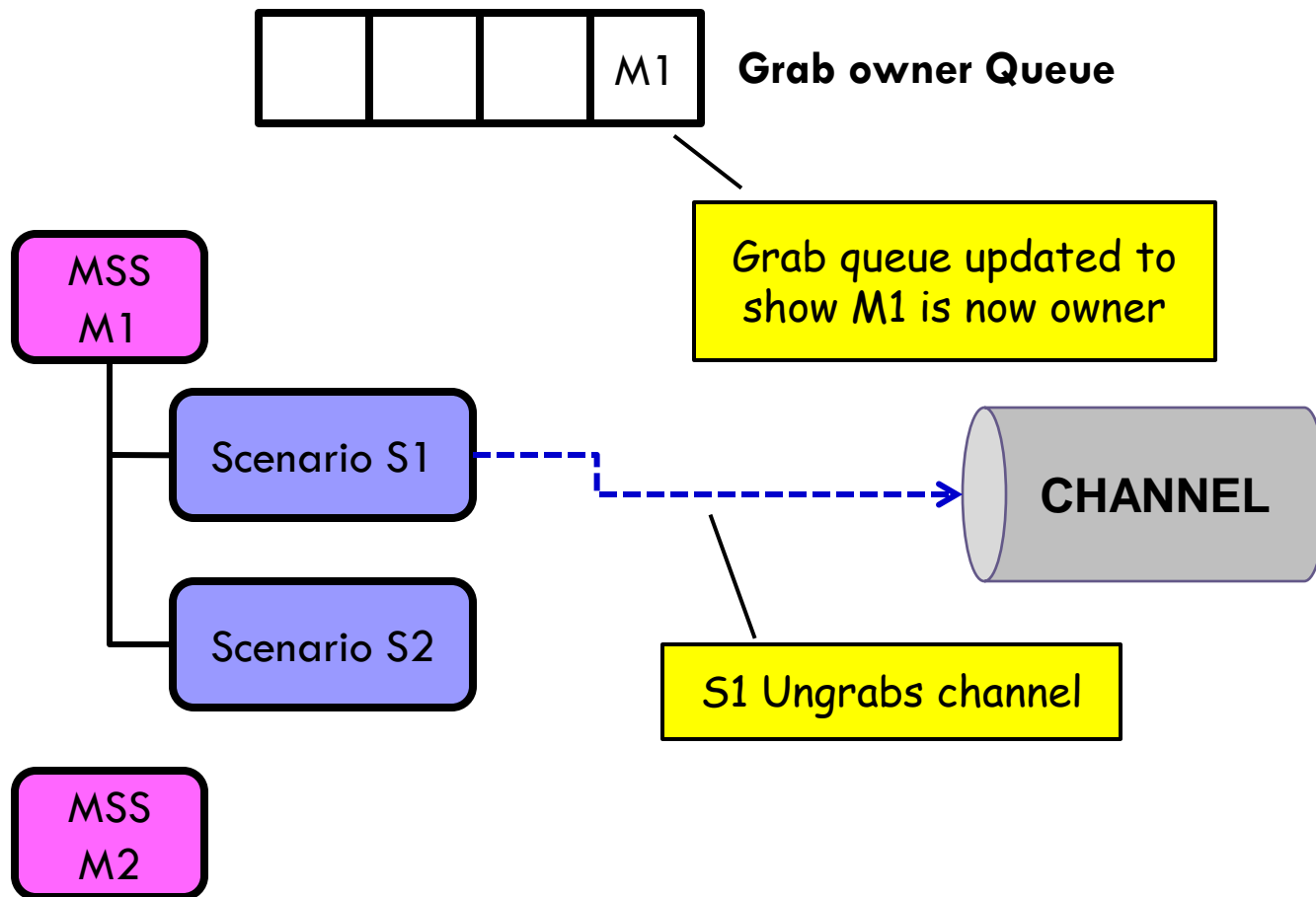
# VMM Channel Grab/Ungrab



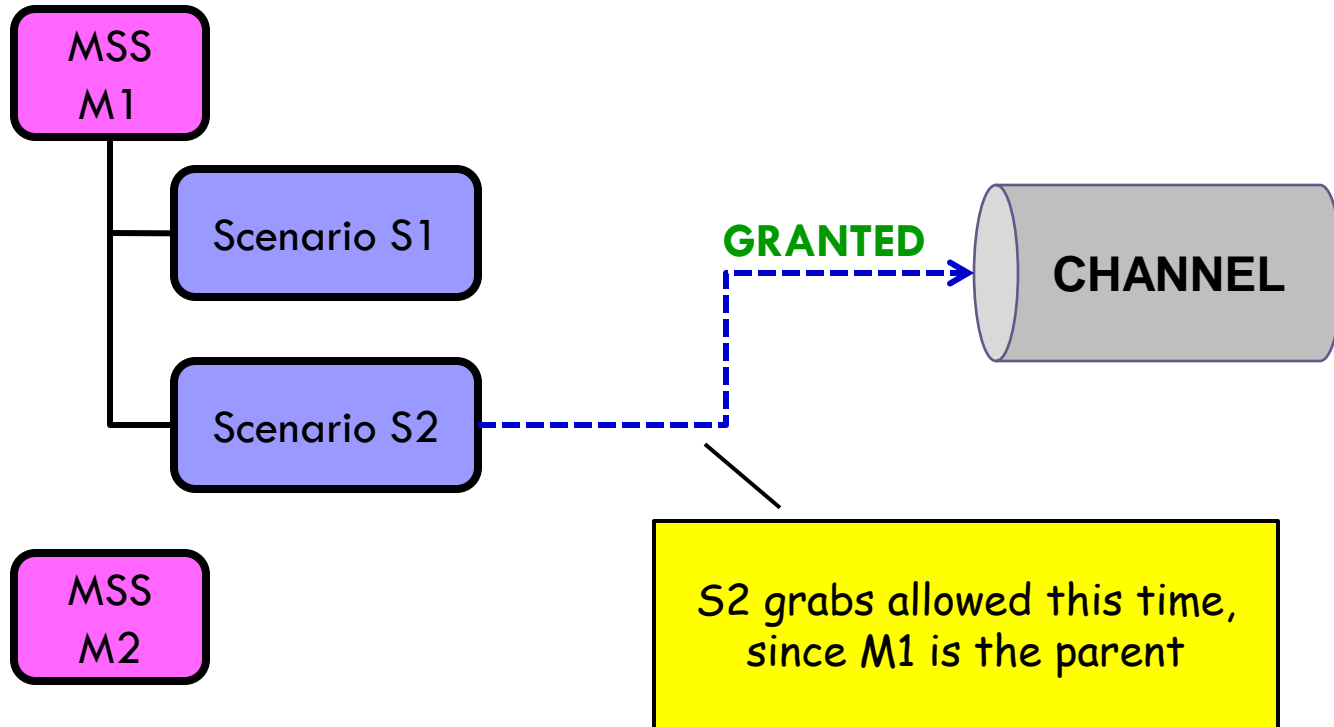
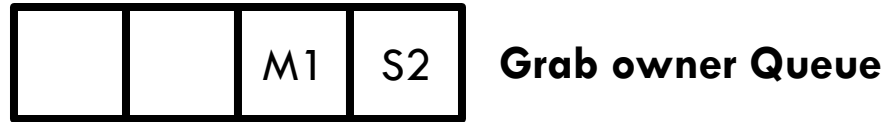
# VMM Channel Grab/Ungrab



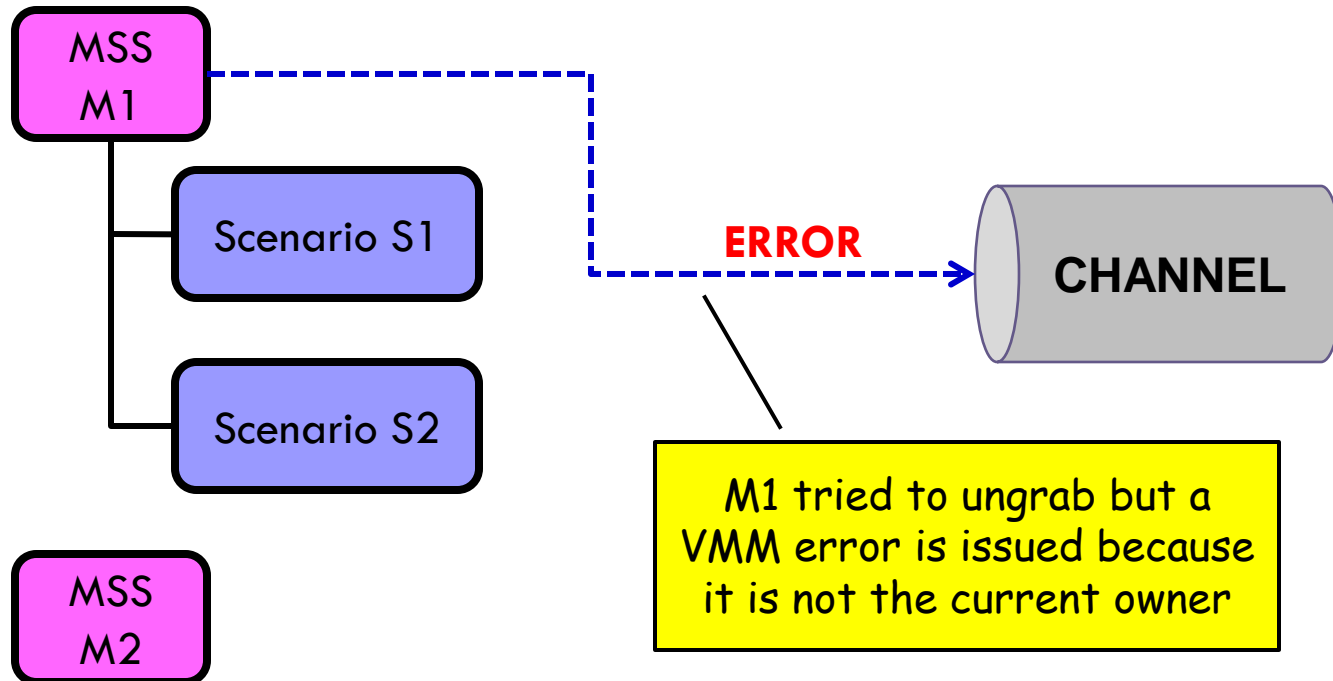
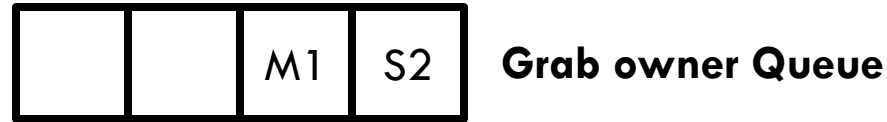
# VMM Channel Grab/Ungrab



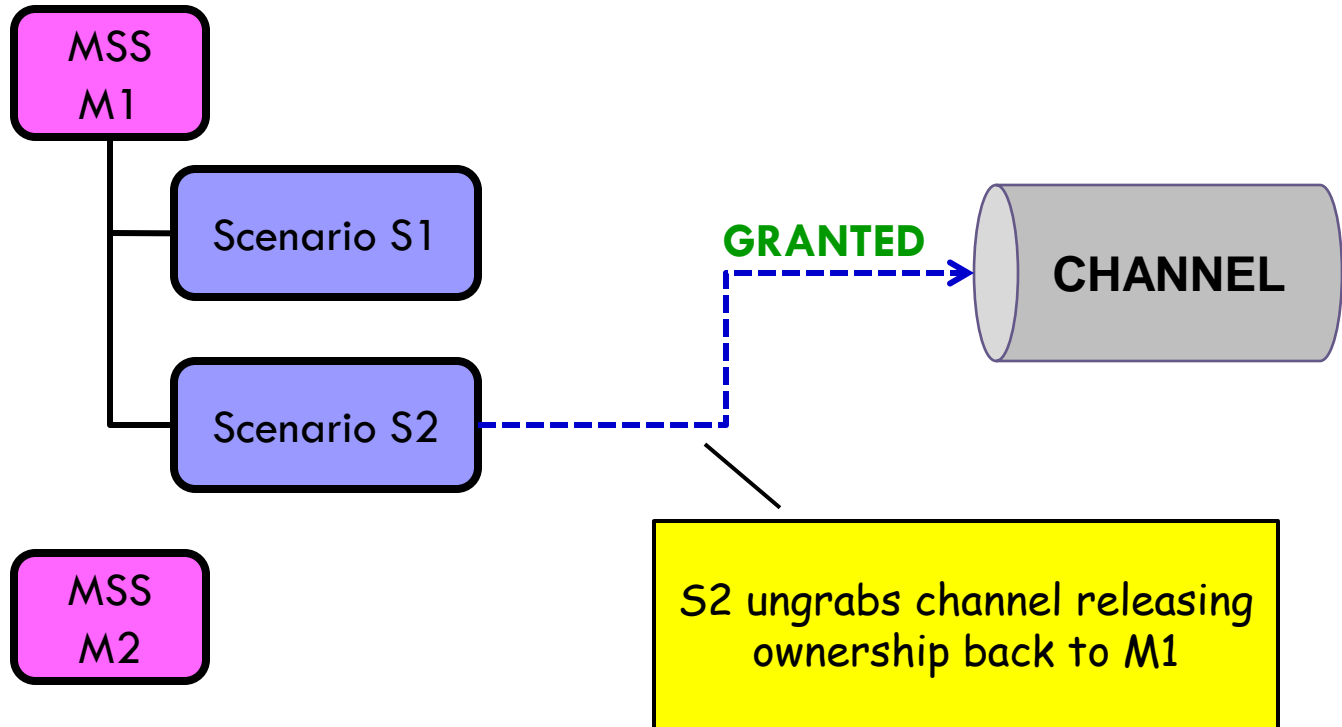
# VMM Channel Grab/Ungrab



# VMM Channel Grab/Ungrab

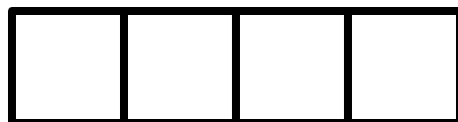


# VMM Channel Grab/Ungrab

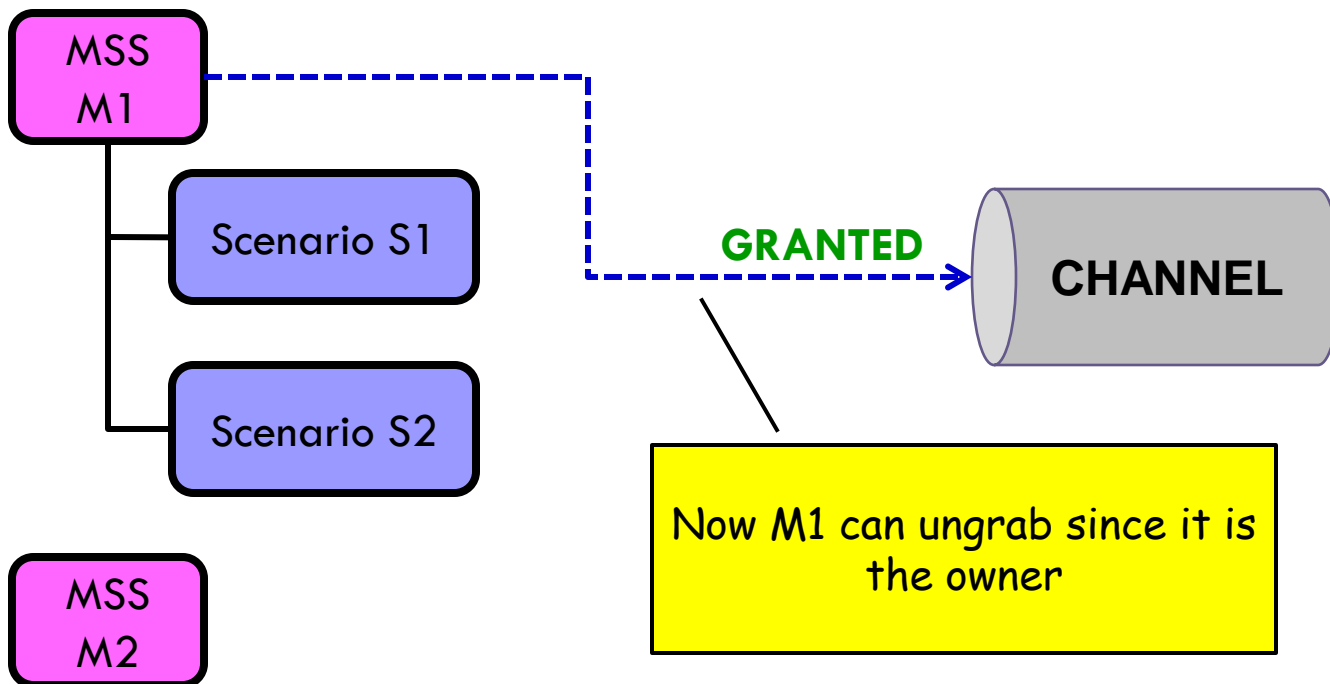




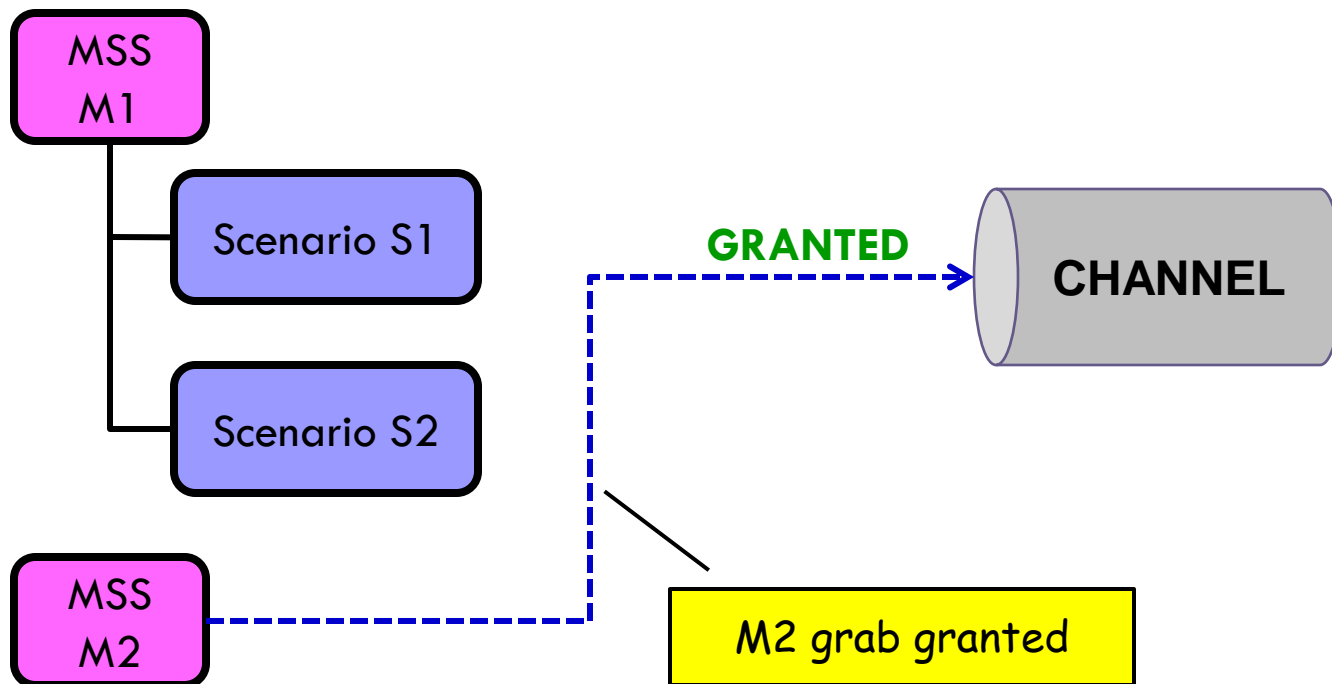
# VMM Channel Grab/Ungrab



Grab owner Queue



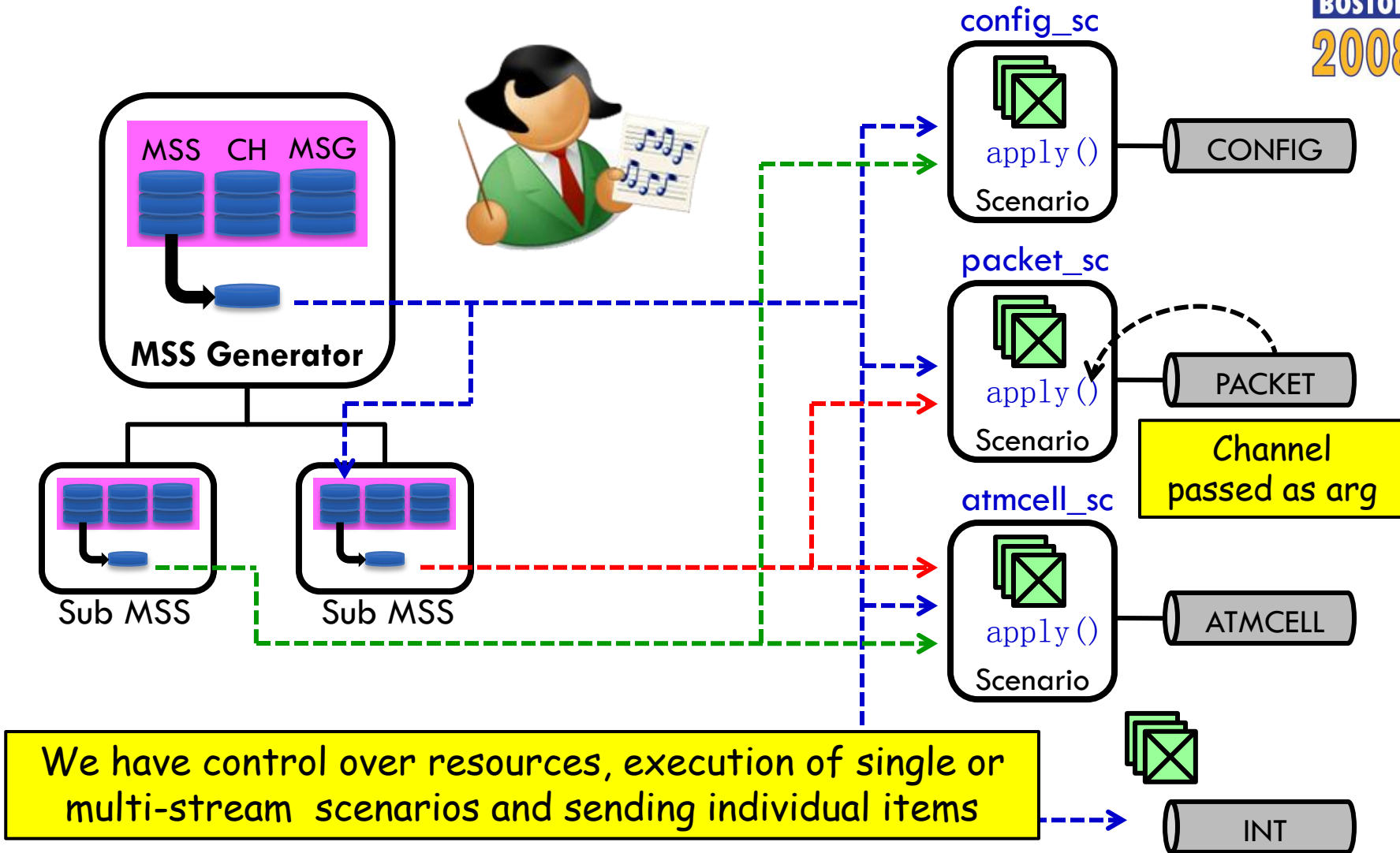
# VMM Channel Grab/Ungrab



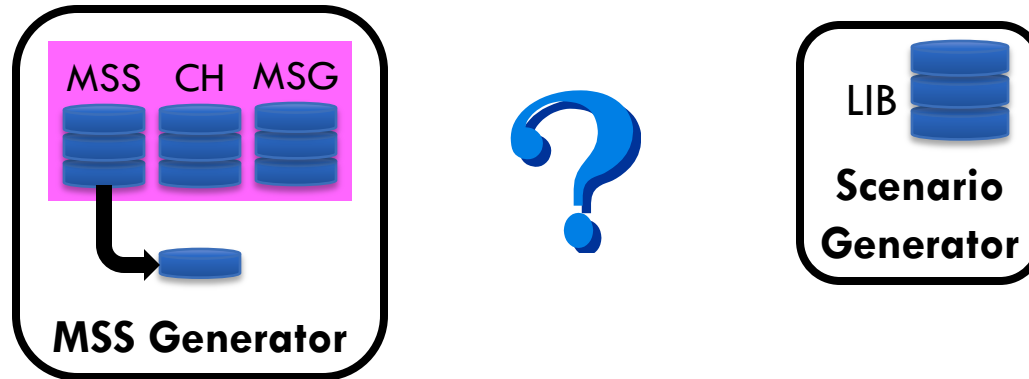
# Grab Features In vmm\_channel

Feature	Description
GRABBED/UNGRABBED events	Notify events
function void reset_grabbers()	Removes all grabs
function bit try_grab(vmm_scenario grabber)	Attempts a grab, returning status (non blocking)
task grab(vmm_scenario grabber)	Tries to grab a channel (blocking)
function void ungrab(vmm_scenario grabber)	Tries to ungrab a channel. Returns a VMM error if illegal (non blocking)
function bit is_grabbed()	Returns 1 if the channel is grabbed

# MSS Overview



# Do MSS Replace Scenarios?

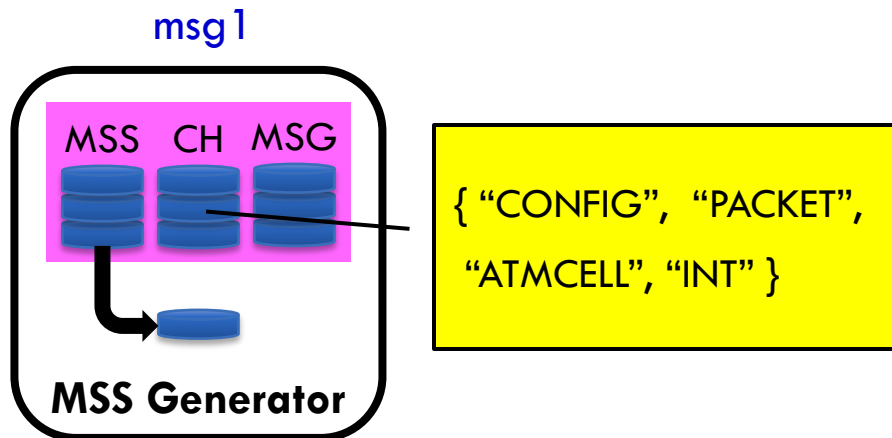


- MSS brings multi-stream control
- MSS can definitely replace scenarios, but ...
- They don't have to – both play together no problem
- MSS Generator doesn't have a registry for single scenarios
- With MSS we don't have some features, e.g. atomic and default scenarios

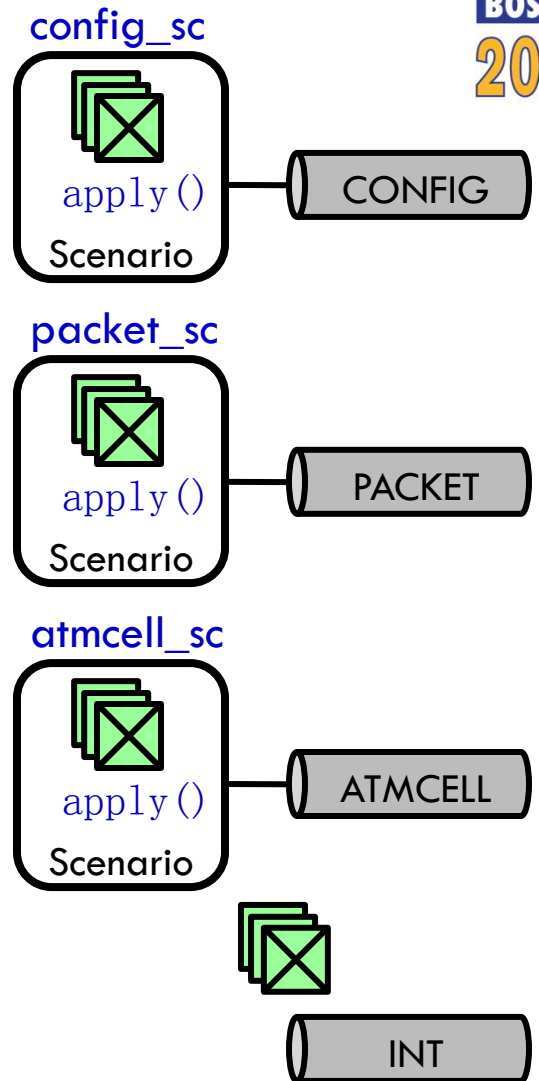
# MSS vs Single Scenario

Single Scenario	Multi-stream Scenario
Communicates with the channel via <code>apply()</code> where channel is passed as an argument	Looks up handle for channel in its channel registry.
Can only call scenarios that are capable of using the same channel.	Can call other scenarios by storing their channel in the channel registry
Cannot call multi-stream scenarios	Can call other multi-stream scenarios in accessible multi-stream scenario generators

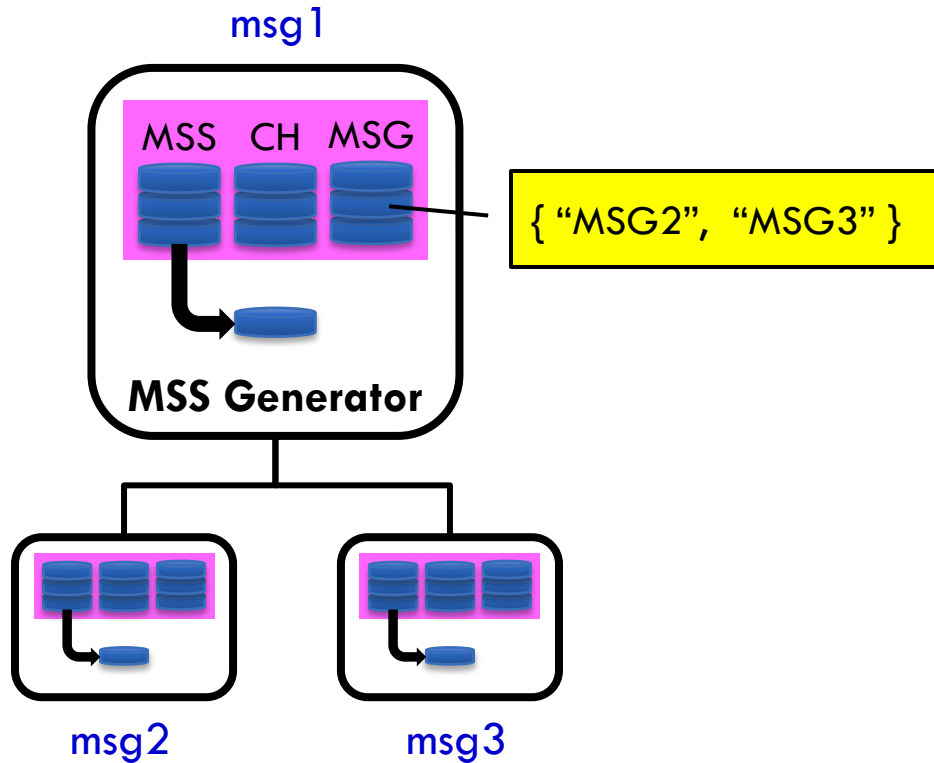
# MSS Channel Registry



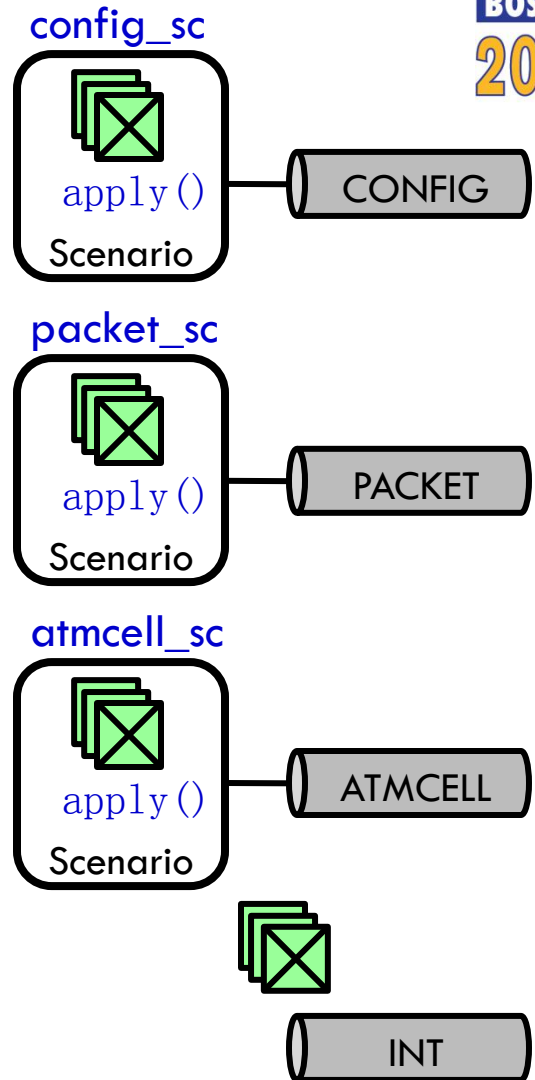
```
// register any channels being used
msg1.register_channel( "CONFIG" , config_ch);
...
msg1.register_channel( "INT" , interrupt_ch);
```



# MSS Generator Registry

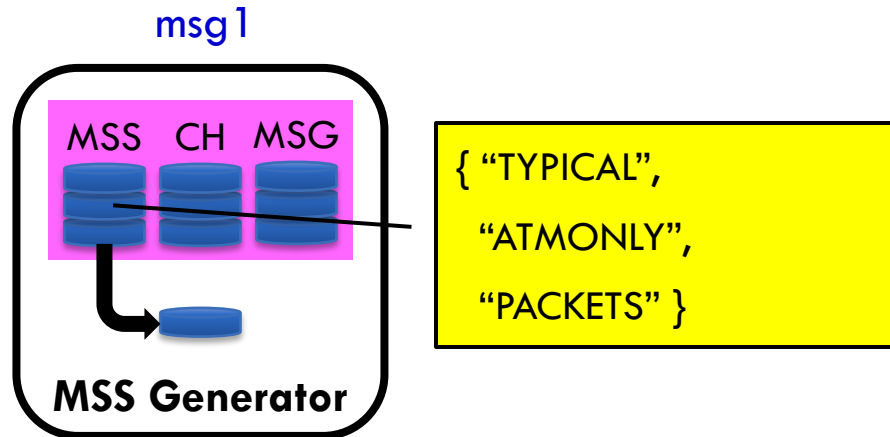


```
// register MSS gens connected
msg1.register_ms_scenario_gen( "MSG2" , msg2);
msg1.register_ms_scenario_gen( "MSG3" , msg3);
```



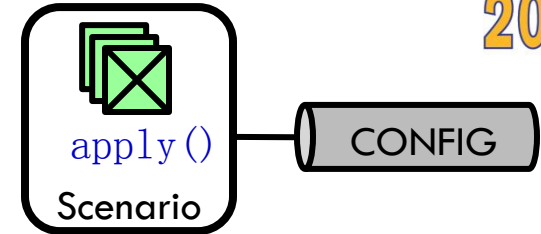


# MSS Registry (Library)

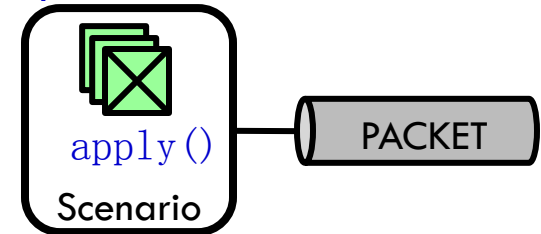


```
// register MSS being used
msg1.register_ms_scenario( "CONFIG" , ms_typical);
msg1.register_ms_scenario( "ATMONLY" , ms_atmonly);
msg1.register_ms_scenario( "PACKETS" , ms_packets);
```

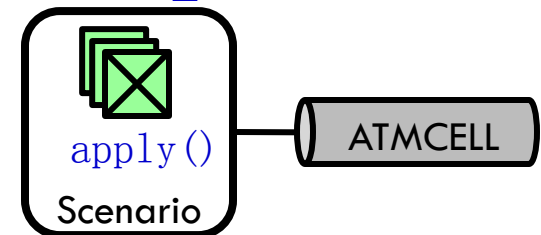
config\_sc



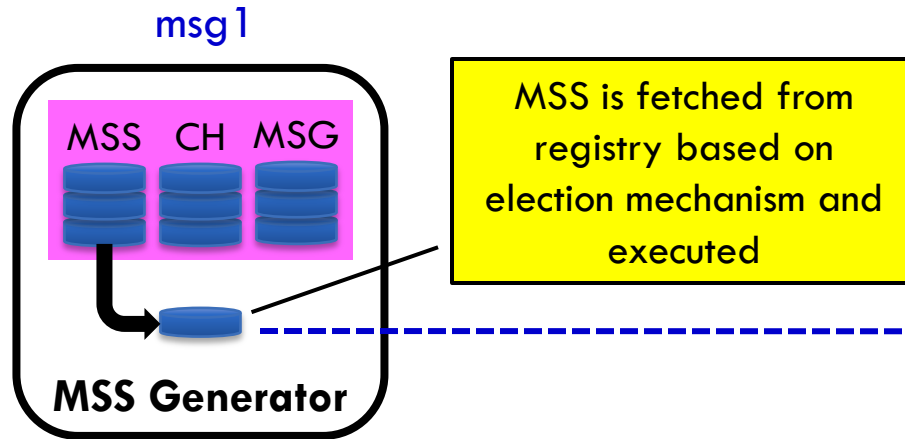
packet\_sc



atmcell\_sc

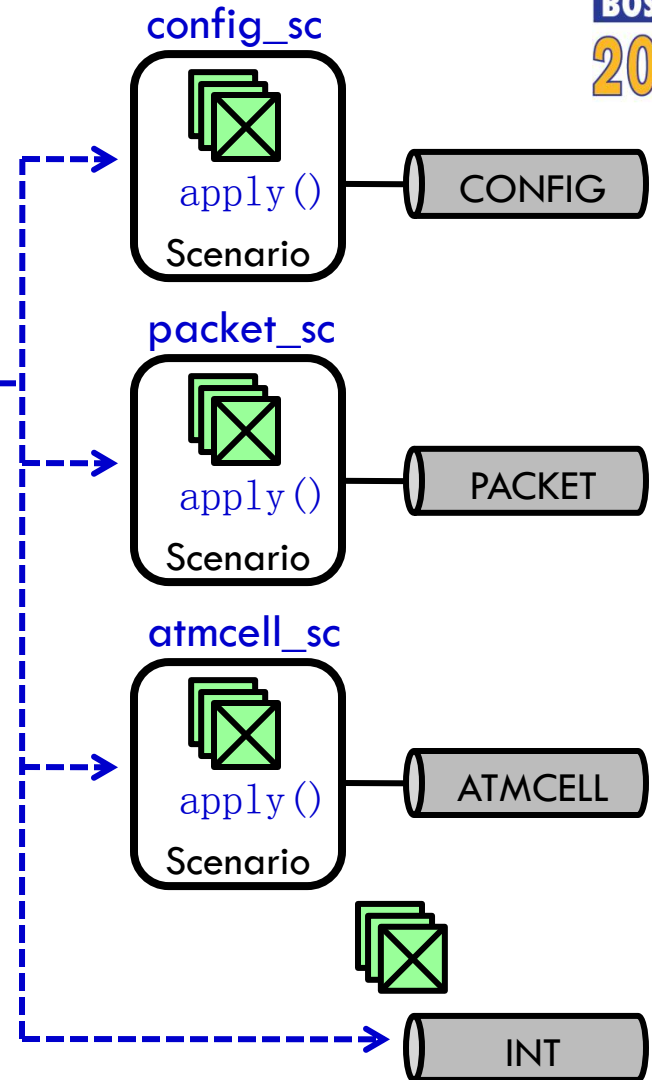


# MSS Execution



```

ms_typical extends vmm_ms_scenario;
...
virtual task execute(ref int n_insts);
    // MSS actions go here
endtask
endclass
    
```



# Walkthrough Of MSS execute()

```
// Extend from the base class
class ms_typical extends vmm_ms_scenario;
. . .
virtual task execute(ref int n_insts);
    int n=0;
    vmm_channel config_chan;    // for single scenario gen
    vmm_channel packet_chan;    // for single scenario gen
    vmm_channel atm_cell_chan;  // for single scenario gen
    vmm_channel interrupt_chan; // for sending items directly
    packet_scenario  packet_sc    = new(this);
    atm_cell_scenario atm_cell_sc  = new(this);
    config_scenario  config_sc    = new(this);
    interrupt_ctl    my_interrupts = new(); // single item
```

Continued ...

# Walkthrough Of MSS execute()

```
// Get the channel handles from registry
config_chan    = this.get_channel( "CONFIG" );
packet_chan    = this.get_channel( "PACKET" );
atm_cell_chan  = this.get_channel( "ATM_CELL" );
interrupt_chan = this.get_channel( "INT" );

// grab the channels we want exclusive access to
config_chan.grab(this);
packet_chan.grab(this);
atm_cell_chan.grab(this);

// Do the configuration scenario first
config_sc.randomize with { ... };
config_sc.apply(config_chan, n);
n_insts++;
```

Continued ...

# Walkthrough Of MSS execute()

```
// Run other scenarios and send items in parallel
fork
  begin
    packet_sc1.randomize();
    packet_sc1.apply(packet_chan, n);
    n_insts++;
  end
  begin
    atm_cell_sc.randomize();
    atm_cell_sc.apply(atm_cell_chan, n);
    n_insts++;
  end
end
```

Continued ...

# Walkthrough Of MSS execute()

```
begin
    my_interrrupts.randomize() with { ... };
    interrupt_ch.put(my_interrrupts);
    n_insts++;
end
join
// Release the channels
config_chan.ungrab(this);
packet_chan.ungrab(this);
atm_cell_chan.ungrab(this);
endtask: execute
. . .
endclass: ms_normal
```

# Summary

- Added functionality to enhance resource locking control
- Added ability to control multiple streams of stimulus
- Augments current scenario generation
- Backward compatible

# Jason Sprott (V.P. Consulting) Verilab

[jason.sprott@verilab.com](mailto:jason.sprott@verilab.com)

[www.verilab.com](http://www.verilab.com)

