



VERIFICATION ACADEMY

Basic OVM

Connecting Env to DUT

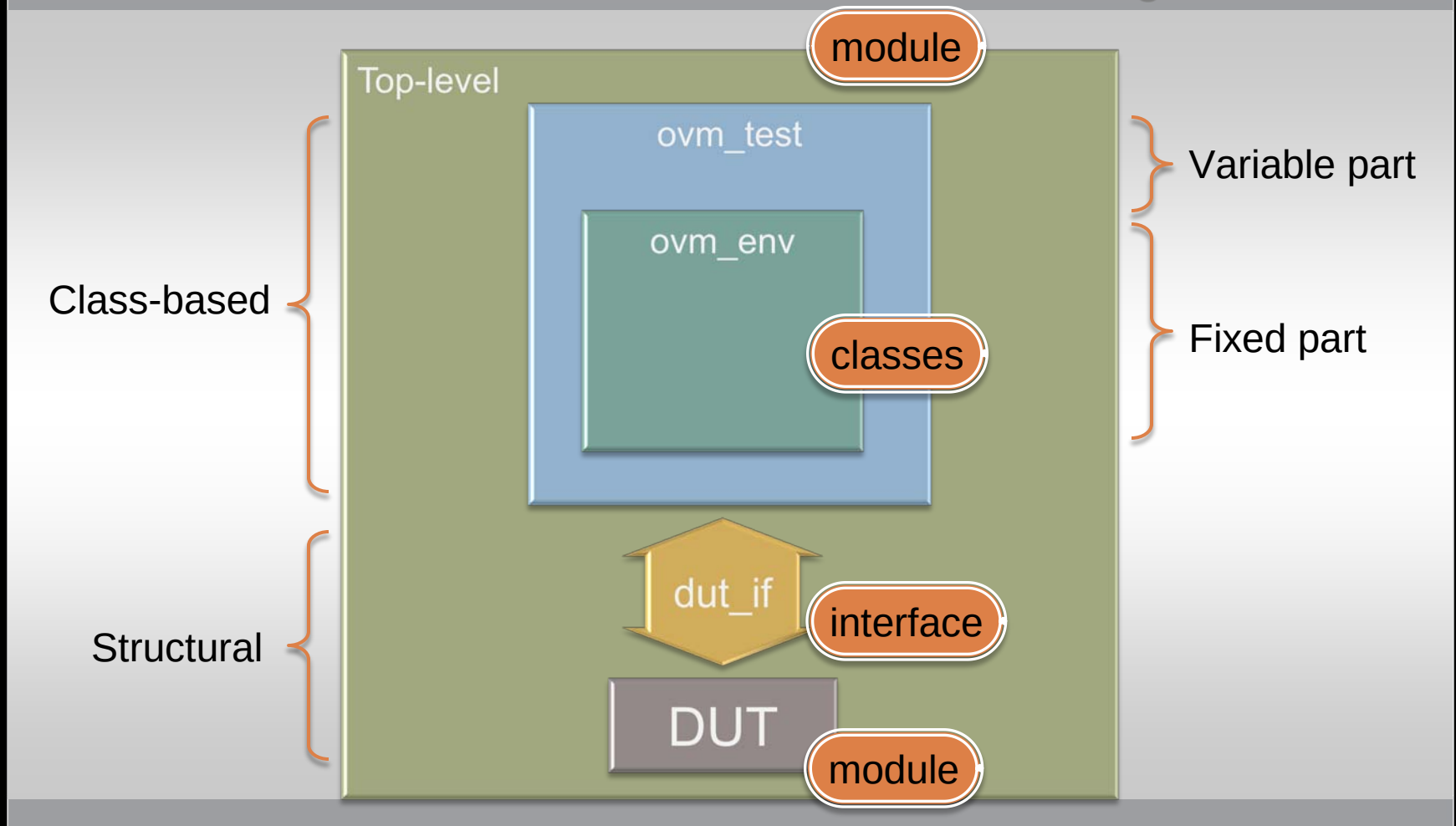
John Aynsley
CTO, Doulos

academy@mentor.com
www.verifacationacademy.com





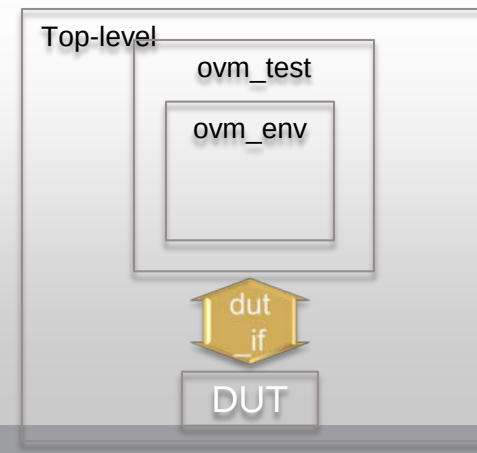
Connecting Env to DUT





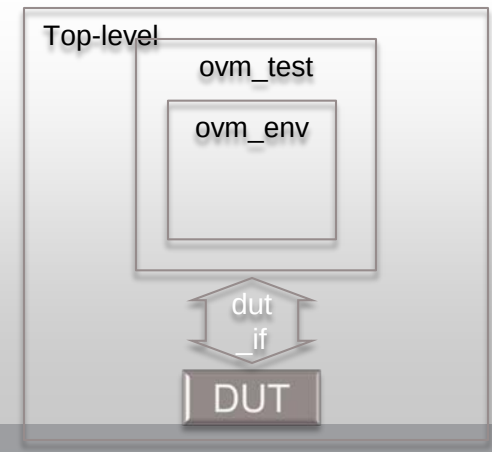
Interface

```
interface dut_if();  
  
    logic clock, reset;  
    logic data;  
  
endinterface: dut_if
```





```
module dut(dut_if _if);  
  
    always @(posedge _if.clock)  
    begin  
        ...  
    end  
  
endmodule: dut
```





Accessing Pins through Virtual Interface

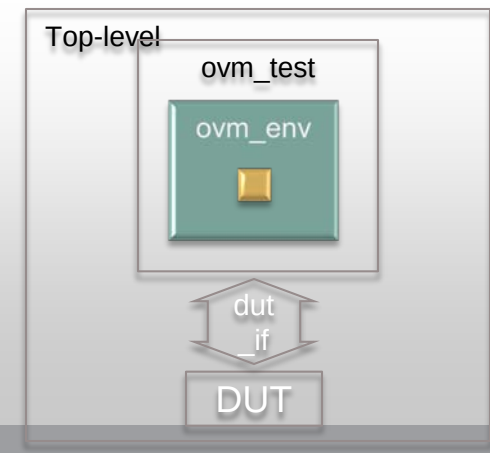
```
class my_env extends ovm_env;  
  `ovm_component_utils(my_env)
```

```
  virtual dut_if dut_vi;
```



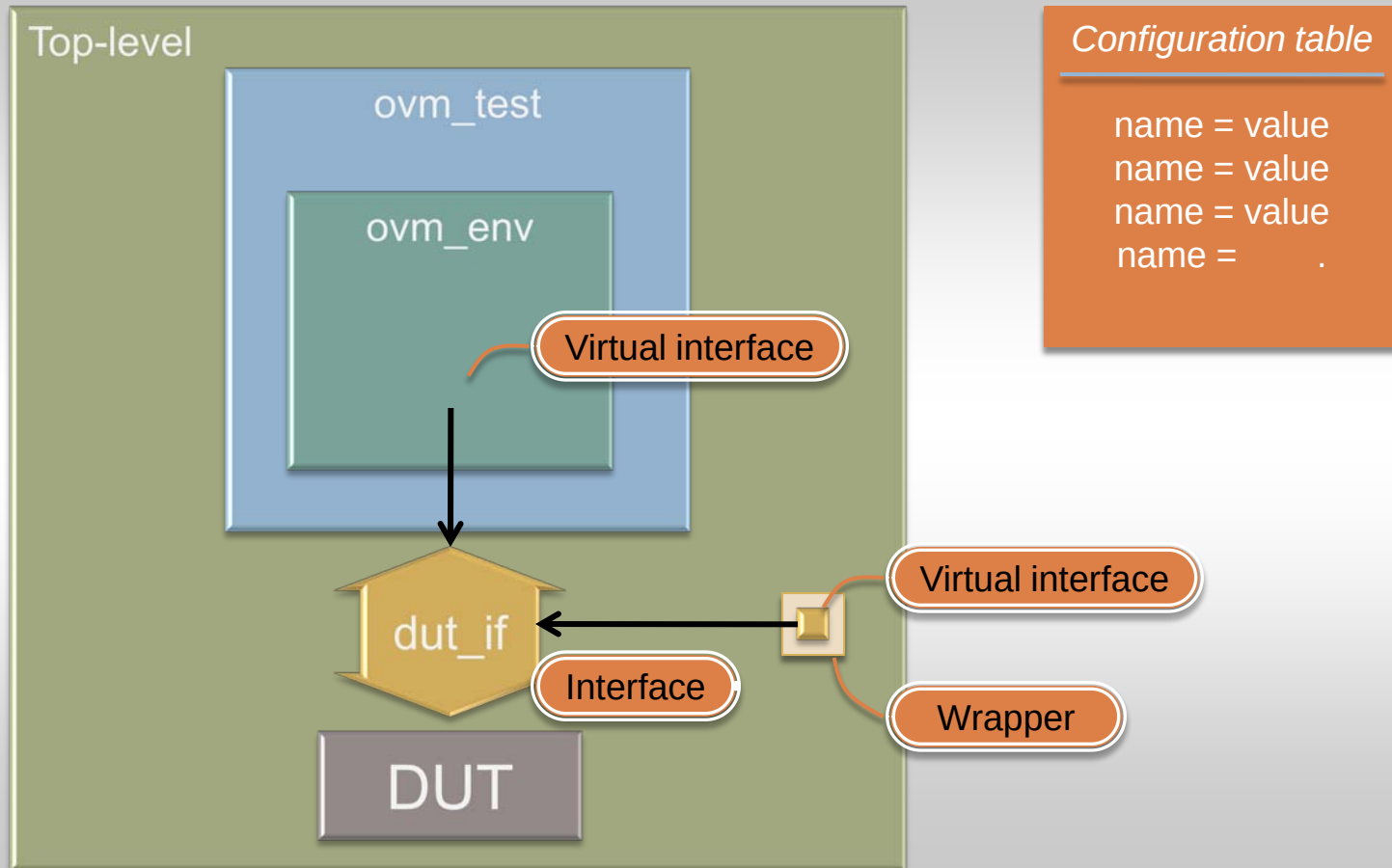
_vi = virtual interface

```
  function new(string name, ovm_component parent);  
    ...  
  function void build;  
    ...  
  
  task run;  
    #10 dut_vi.data = 0;  
    #10 dut_vi.data = 1;  
    #10 ovm_top.stop_request();  
  endtask: run  
  ...
```



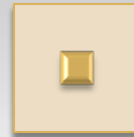


Configuration Table



Virtual Interface Wrapper

```
class dut_if_wrapper extends ovm_object;  
  
    virtual dut_if dut_vi;  
  
    function new(string name, virtual dut_if arg);  
        super.new(name);  
        dut_vi = arg;  
    endfunction: new  
  
endclass: dut_if_wrapper
```



set_config_object

```
module top;  
  ...  
  dut_if dut_if1 ();  
  
  initial  
  begin: blk  
    dut_if_wrapper if_wrapper = new("if_wrapper", dut_if1);
```



set_config_object

```
module top;  
  ...  
  dut_if dut_if1 ();  
  
  initial  
  begin: blk  
    dut_if_wrapper if_wrapper = new("if_wrapper", dut_if1);  
  
    set_config_object("*", "dut_if_wrapper", if_wrapper, 0);
```

PathField nameValueDon't clone

set_config_object

```
module top;
  ...
  dut_if dut_if1 ();

  initial
  begin: blk
    dut_if_wrapper if_wrapper = new("if_wrapper", dut_if1);

    set_config_object("*", "dut_if_wrapper", if_wrapper, 0);

    run_test("my_test");
  end

endmodule: top
```

get_config_object

```
class my_env extends ovm_env;
...
virtual dut_if dut_vi;
...

function void build;
    super.build();
    begin
        ovm_object obj;
        dut_if_wrapper if_wrapper;
        get_config_object("dut_if_wrapper", obj, 0);
```

Field name

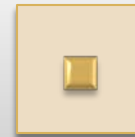
Value

Don't clone

get_config_object

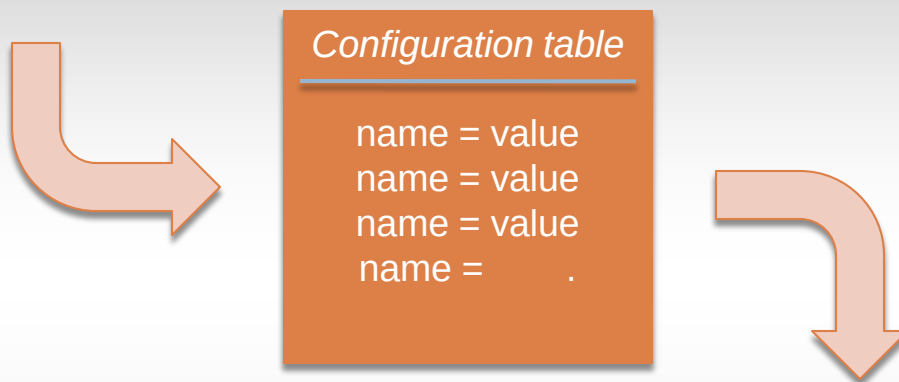
```
class my_env extends ovm_env;
...
virtual dut_if dut_vi;
...

function void build;
    super.build();
    begin
        ovm_object obj;
        dut_if_wrapper if_wrapper;
        get_config_object("dut_if_wrapper", obj, 0);
        assert( $cast(if_wrapper, obj) );
        dut_vi = if_wrapper.dut_vi;
    end
endfunction: build
```



Setting and Getting Configuration Values

```
set_config_object("path", "name", value, 0);
```



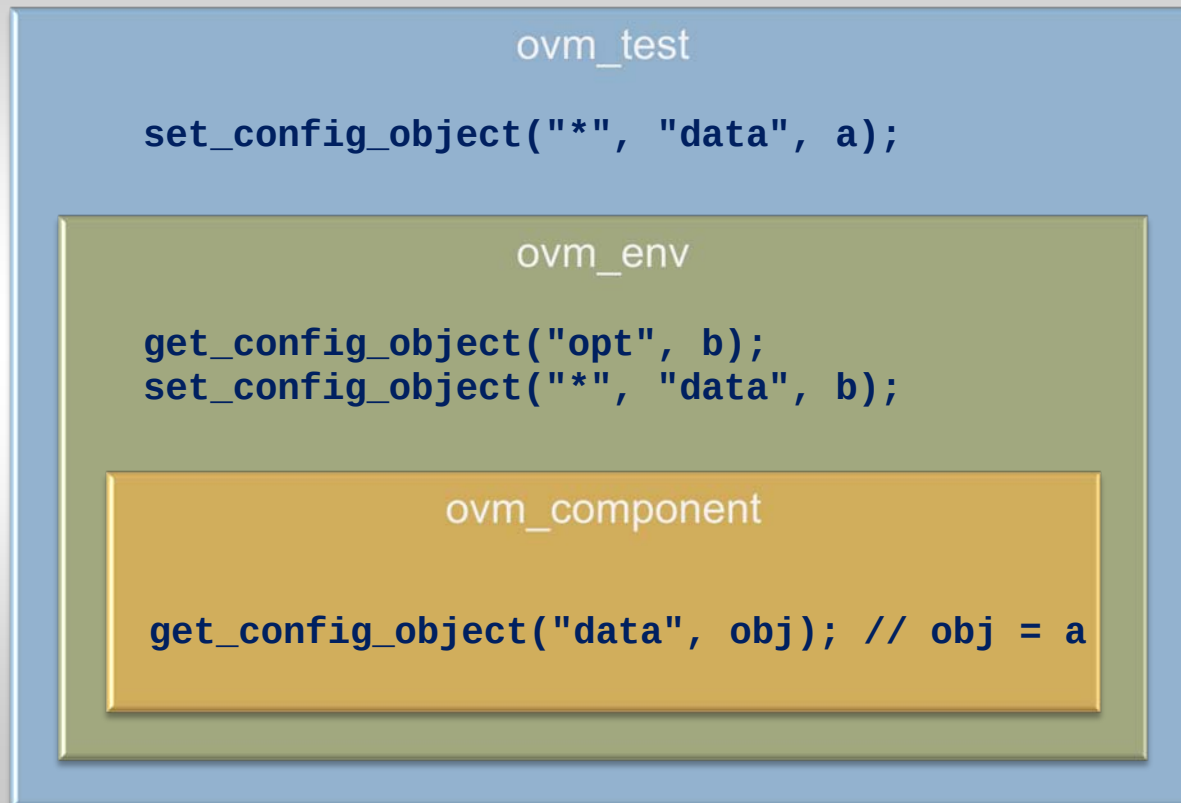
```
get_config_object("name", value, 0);
```

Path and field names can contain wildcards



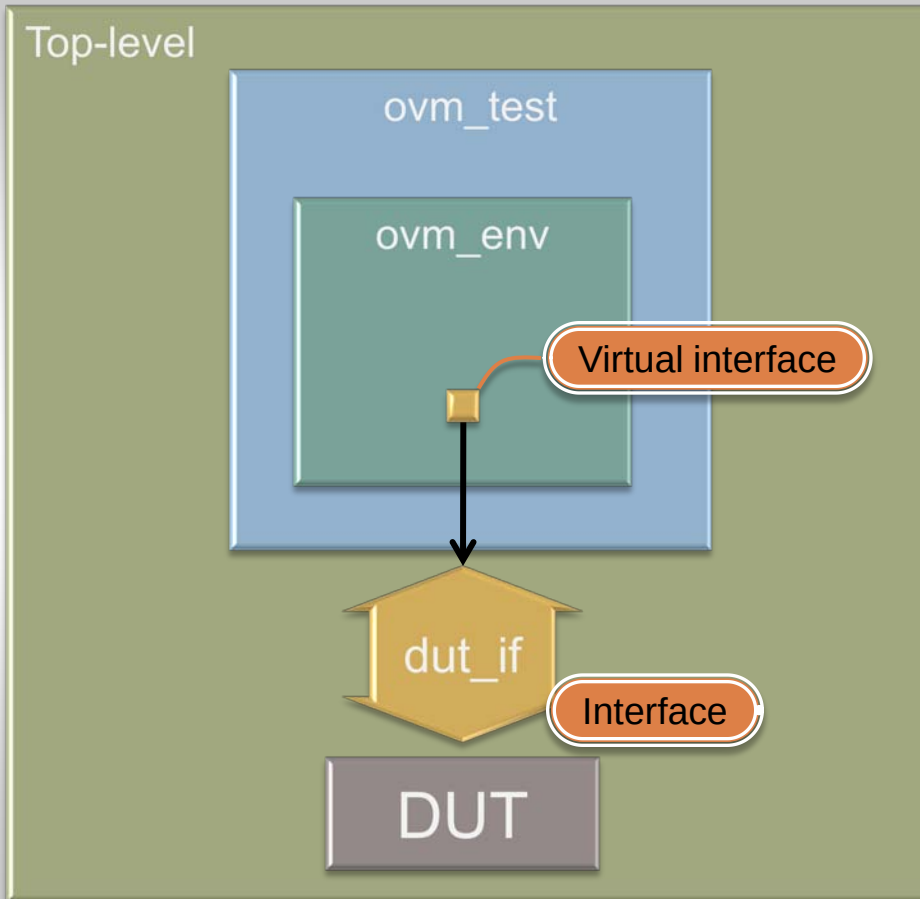
Hierarchical Configuration

Config information flows top-down, higher levels overriding lower levels





Summary



Configuration table

name = value
name = value
name = value
name = 



VERIFICATION ACADEMY

Basic OVM

Connecting Env to DUT

John Aynsley
CTO, Doulos

academy@mentor.com
www.verifacationacademy.com

