Sistemi Lente/Prism

A Manufacturing Test Framework

The (good) Problem…

- Startup/Small Company develops a product, builds a prototype…
  - Customer loves it, orders thousands, due in 3 months…
  - Time to build a test system…
- Can you afford/budget to outsource the development?
  - Did you write clean specs to transfer knowledge of your product to an outsource to get the job done (in time)?
  - Can your core developers support a 3rd party while they prepare for launch?
- Can you develop the test system yourself?
  - More software, another PCB design…
  - Most test systems are more/as complicated as the product they test…
    - User Interface, Database schema, circuit performance/debug/bringup, revision control, dashboarding, security, deployment, …

Why do this in-house?

- Most HW companies do…
- Flexibility (Change Management)
  - Product design changes often domino into the production test system
    - change in limits
    - change in sequence
    - new test, delete old test
    - etc
  - Always going to be FASTER to do this internally rather than externally (CM)
- CM Freedom
  - Not tied to the CM system
- PCBA Testing is just one step,
  - If the PCBA is integrated into a product, with other devices, a final test is probably required and that is planned to be done internally
- Test System becomes company IP & Competitive Advantage

Sistemi Lente/Prism Test Platform

A Framework to Develop/Deploy Production Test Suites

- Graphical (web) User Interface
- JSON style "Scripts" for Test Flow, Limits, etc
- Tests programmed in Python and Arduino
- Production Monitoring Dashboard
- Structured Database schema
- Scalability & Security
- Deployment Strategy and Version Control
- Barcode Travellers for zero-effort/error-free Test Configuration
- User Defined Production Tracking Variables
- Open Source Hardware with Software to get started quickly
- Online Documentation and Examples

Estimated to **save 6-12 man-months of development** for a decent manufacturing test system that has comparable features

Sistemi Lente/Prism Test Platform

THE GOALS ARE

- **TO FREE YOUR TECHNICAL PEOPLE TO WRITE PYTHON TEST SCRIPTS THAT ACTUALLY TEST YOUR PRODUCT.**
- **YOUR TEST SYSTEM SHOULD NOT CONTRIBUTE TO YOUR TECHNICAL DEBT.**

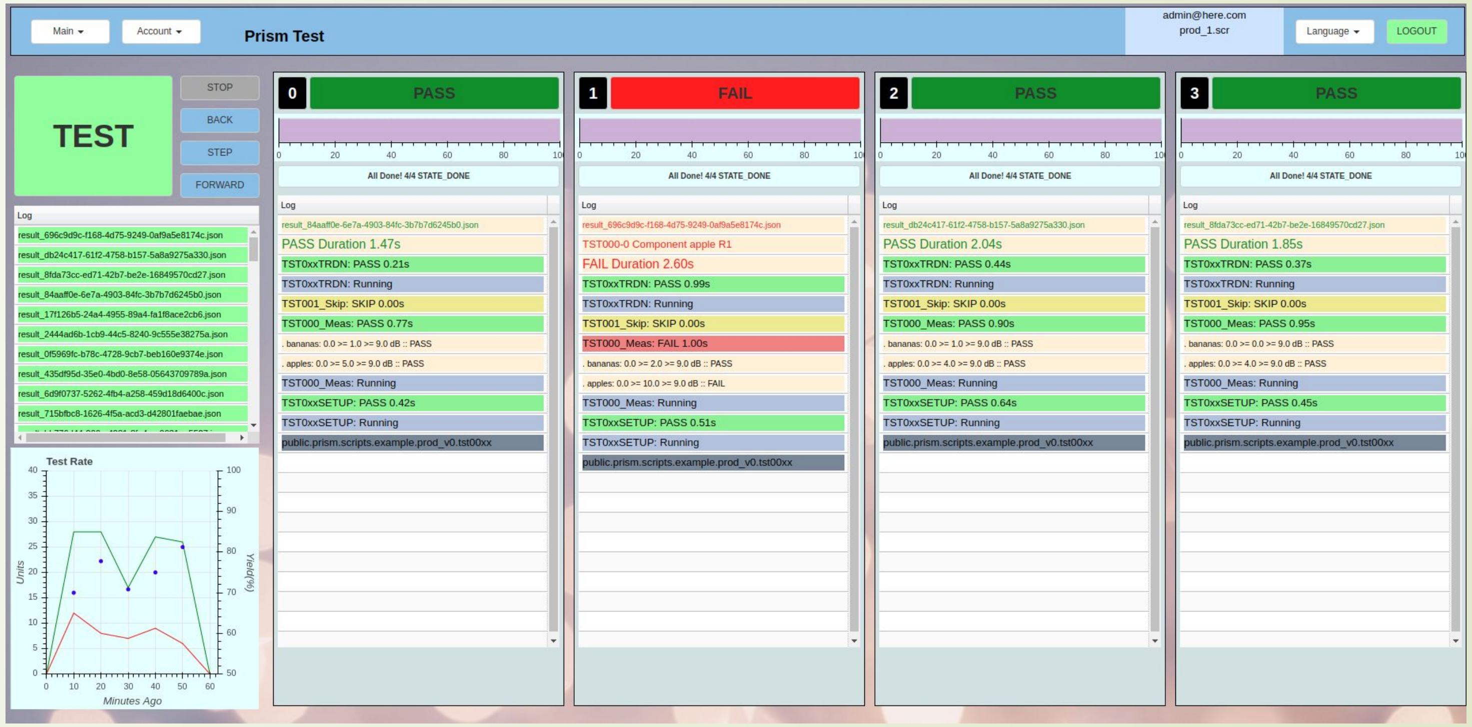
Sistemi Lente/Prism Test Platform

SCREENSHOTS AND HIGH LEVEL TECHNICAL DETAILS

Sistemi Lente/Prism Test Platform

Graphical User Interface

- ➔ Color coded eye-catching views
- ➔ Get key metrics quickly
- ➔ Supports Multiple Languages



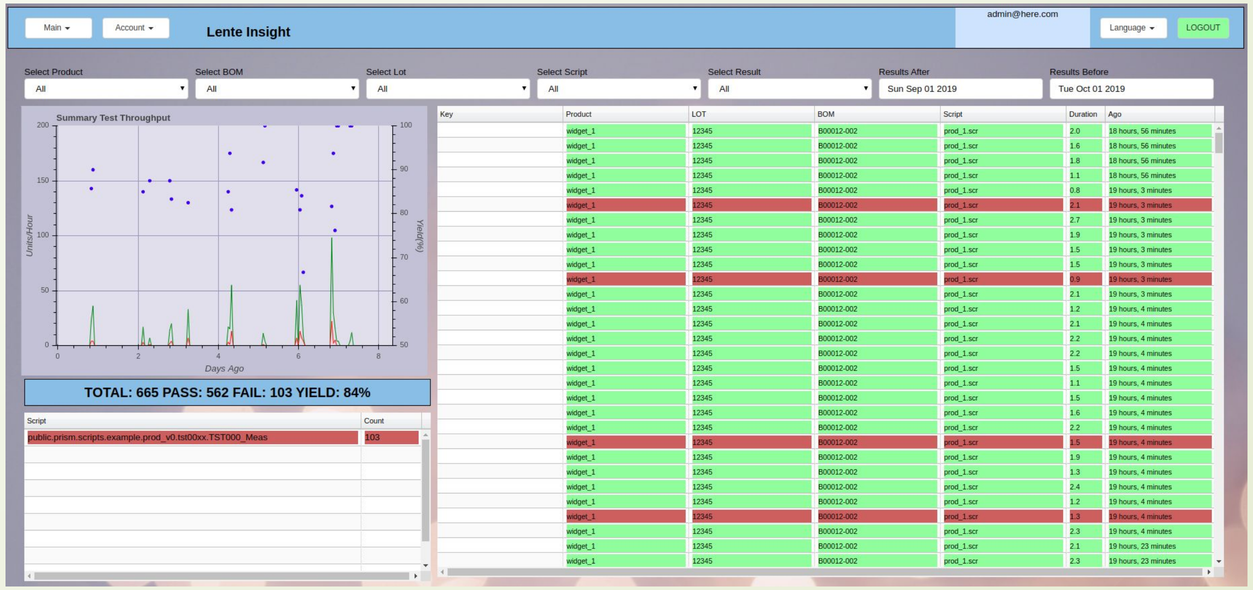
```
Log
result_696c9d9c-f168-4d75-9249-0af9a5e8174c.json
result_db24c417-61f2-4758-b157-5a8a9275a330.json
result_8fda73cc-ed71-42b7-be2e-16849570cd27.json
result_84aaff0e-6e7a-4903-84fc-3b7b7d6245b0.json
result_17f126b5-24a4-4955-89a4-fa1f8ace2cb6.json
result_2444ad6b-1cb9-44c5-8240-9c555e38275a.json
result_0f5969fc-b78c-4728-9cb7-beb160e9374e.json
result_435df95d-35e0-4bd0-8e58-05643709789a.json
result_6d9f0737-5262-4fb4-a258-459d18d6400c.json
result_715bfbc8-1626-4f5a-acd3-d42801faebae.json
```

```
0  PASS
All Done! 4/4 STATE_DONE
Log
result_84aaff0e-6e7a-4903-84fc-3b7b7d6245b0.json
PASS Duration 1.47s
TST0xxTRDN: PASS 0.21s
TST0xxTRDN: Running
TST001_Skip: SKIP 0.00s
TST000_Meas: PASS 0.77s
. bananas: 0.0 >= 1.0 >= 9.0 dB :: PASS
. apples: 0.0 >= 5.0 >= 9.0 dB :: PASS
TST000_Meas: Running
TST0xxSETUP: PASS 0.42s
TST0xxSETUP: Running
public.prism.scripts.example.prod_v0.tst00xx
```

```
1  FAIL
All Done! 4/4 STATE_DONE
Log
result_696c9d9c-f168-4d75-9249-0af9a5e8174c.json
TST000-0 Component apple R1
FAIL Duration 2.60s
TST0xxTRDN: PASS 0.99s
TST0xxTRDN: Running
TST001_Skip: SKIP 0.00s
TST000_Meas: FAIL 1.00s
. bananas: 0.0 >= 2.0 >= 9.0 dB :: PASS
. apples: 0.0 >= 10.0 >= 9.0 dB :: FAIL
TST000_Meas: Running
TST0xxSETUP: PASS 0.51s
TST0xxSETUP: Running
public.prism.scripts.example.prod_v0.tst00xx
```

```
2  PASS
All Done! 4/4 STATE_DONE
Log
result_db24c417-61f2-4758-b157-5a8a9275a330.json
PASS Duration 2.04s
TST0xxTRDN: PASS 0.44s
TST0xxTRDN: Running
TST001_Skip: SKIP 0.00s
TST000_Meas: PASS 0.90s
. bananas: 0.0 >= 1.0 >= 9.0 dB :: PASS
. apples: 0.0 >= 4.0 >= 9.0 dB :: PASS
TST000_Meas: Running
TST0xxSETUP: PASS 0.64s
TST0xxSETUP: Running
public.prism.scripts.example.prod_v0.tst00xx
```

```
3  PASS
All Done! 4/4 STATE_DONE
Log
result_8fda73cc-ed71-42b7-be2e-16849570cd27.json
PASS Duration 1.85s
TST0xxTRDN: PASS 0.37s
TST0xxTRDN: Running
TST001_Skip: SKIP 0.00s
TST000_Meas: PASS 0.95s
. bananas: 0.0 >= 0.0 >= 9.0 dB :: PASS
. apples: 0.0 >= 4.0 >= 9.0 dB :: PASS
TST000_Meas: Running
TST0xxSETUP: PASS 0.45s
TST0xxSETUP: Running
public.prism.scripts.example.prod_v0.tst00xx
```

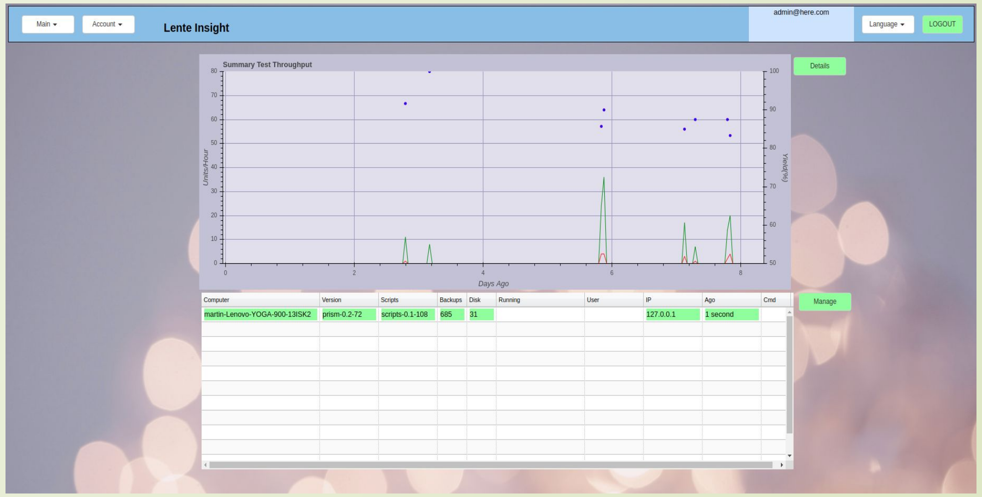


Select Product	Select BOM	Select Lot	Select Script	Select Result	Results After	Results Before
All	All	All	All	All	Sun Sep 01 2019	Tue Oct 01 2019

TOTAL: 665 PASS: 562 FAIL: 103 YIELD: 84%

Script	Count
public.prism.scripts.example.prod_v0.tst00xx.TST000_Meas	103

Key	Product	LOT	BOM	Script	Duration	Ago
	widget_1	12345	B00012-002	prod_1.scr	2.0	18 hours, 56 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.6	18 hours, 56 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.8	18 hours, 56 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.1	18 hours, 56 minutes
	widget_1	12345	B00012-002	prod_1.scr	0.8	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.1	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.7	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.9	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.5	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.5	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	0.9	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.1	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.2	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.1	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.2	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.2	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.5	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.1	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.5	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.6	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.2	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.5	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.9	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.3	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.4	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.2	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.3	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.3	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.1	19 hours, 23 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.3	19 hours, 23 minutes

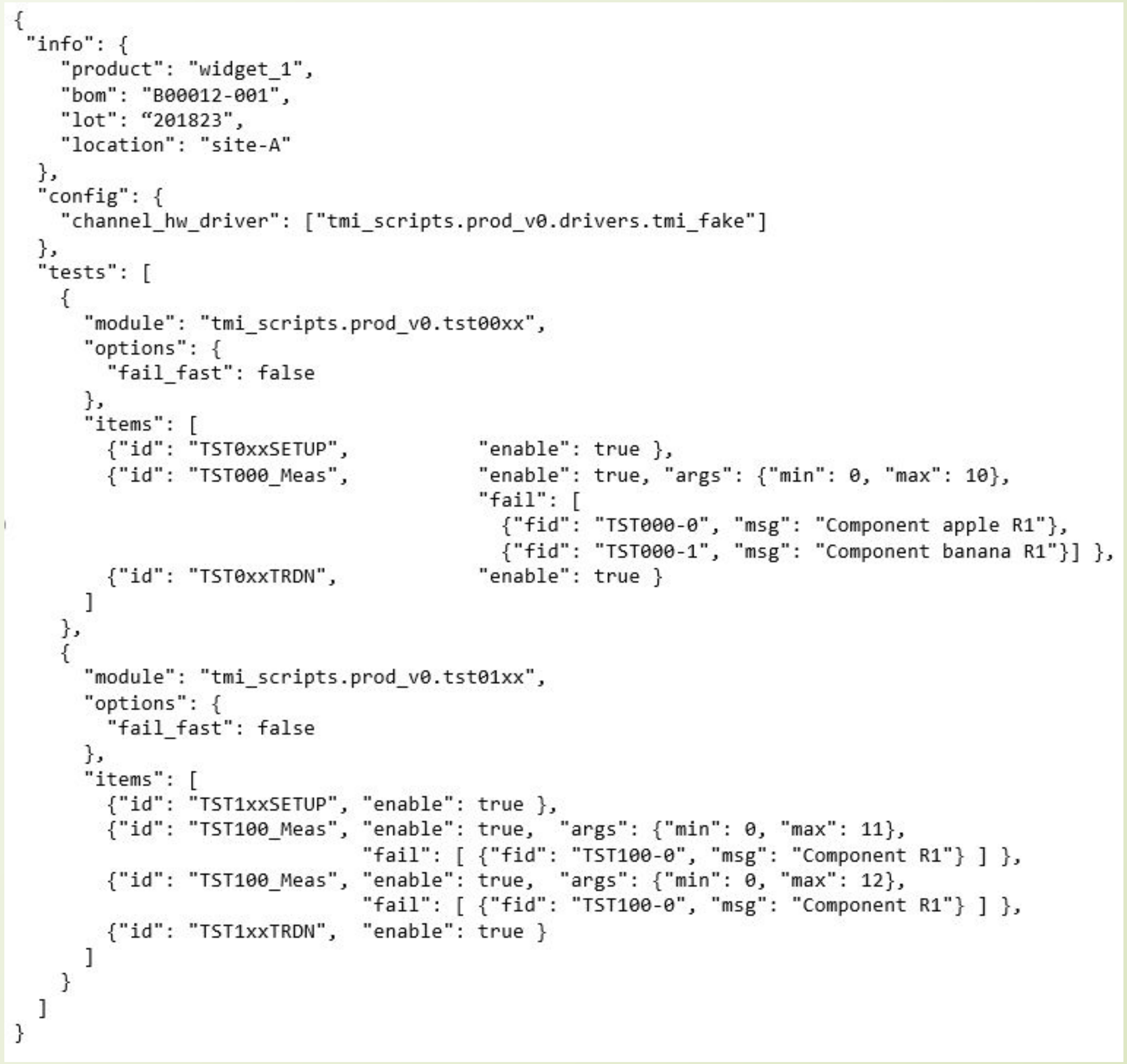


Computer	Version	Scripts	Backups	Disk	Running	User	IP	Ago	Cmd
martin-Lenovo-YOGA-900-13ISK2	prism-0.2-72	scripts-0.1-108	685	31			127.0.0.1	1 second	

Sistemi Lente/Prism Test Platform

JSON Style Test Scripts

- → Drives the test bench
- → Human readable
- → Non-programmer can read this file and make changes
- → Support for GUI driven variable substitution – see Appendix

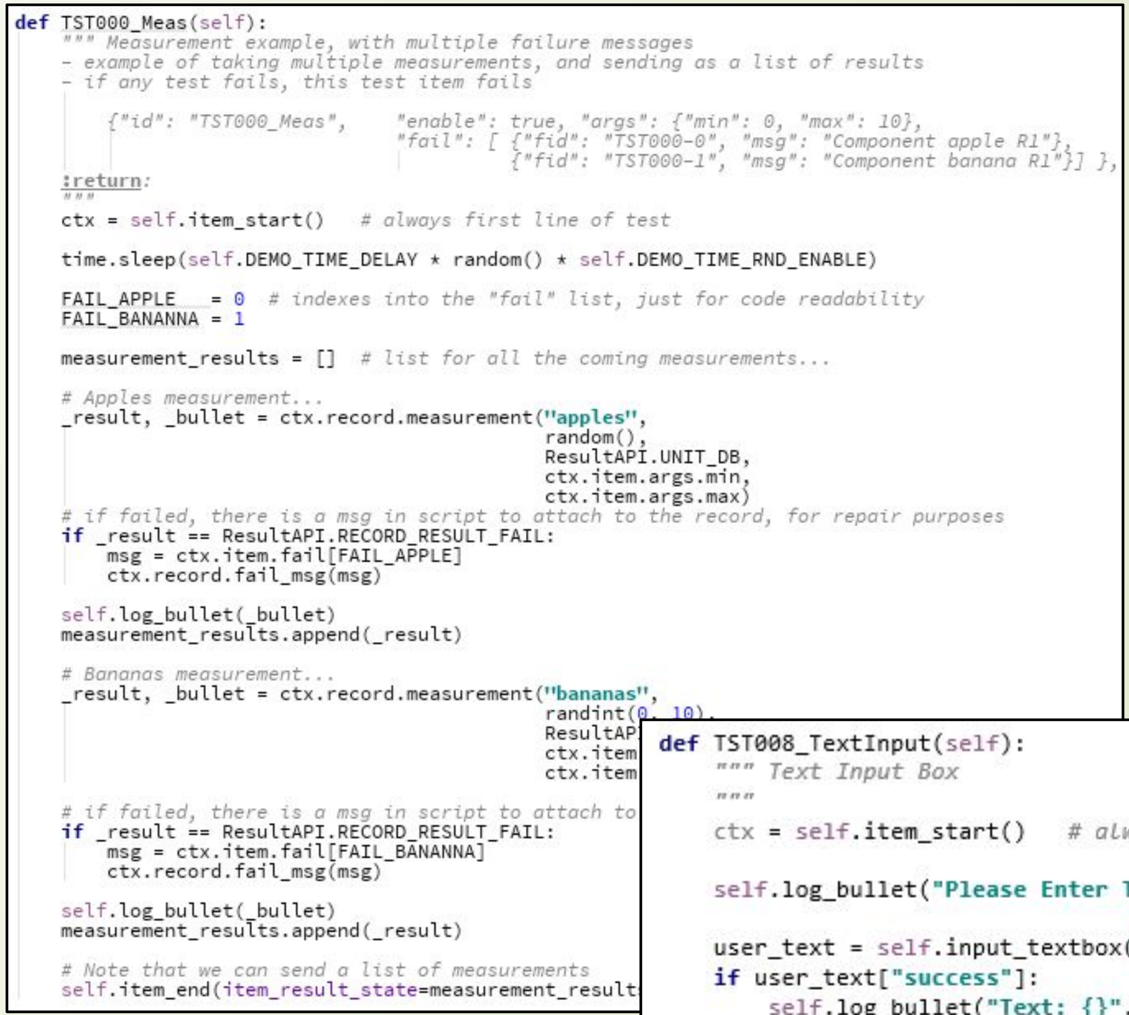


```
{
 "info": {
    "product": "widget_1",
    "bom": "B00012-001",
    "lot": “201823",
    "location": "site-A"
  },
  "config": {
    "channel_hw_driver": ["tmi_scripts.prod_v0.drivers.tmi_fake"]
  },
  "tests": [
    {
      "module": "tmi_scripts.prod_v0.tst00xx",
      "options": {
        "fail_fast": false
      },
      "items": [
        {"id": "TST0xxSETUP",           "enable": true },
        {"id": "TST000_Meas",           "enable": true, "args": {"min": 0, "max": 10},
                                        "fail": [
                                          {"fid": "TST000-0", "msg": "Component apple R1"},
                                          {"fid": "TST000-1", "msg": "Component banana R1"}] },
        {"id": "TST0xxTRDN",            "enable": true }
      ]
    },
    {
      "module": "tmi_scripts.prod_v0.tst01xx",
      "options": {
        "fail_fast": false
      },
      "items": [
        {"id": "TST1xxSETUP", "enable": true },
        {"id": "TST100_Meas", "enable": true,  "args": {"min": 0, "max": 11},
                              "fail": [ {"fid": "TST100-0", "msg": "Component R1"} ] },
        {"id": "TST100_Meas", "enable": true,  "args": {"min": 0, "max": 12},
                              "fail": [ {"fid": "TST100-0", "msg": "Component R1"} ] },
        {"id": "TST1xxTRDN",  "enable": true }
      ]
    }
  ]
}
```

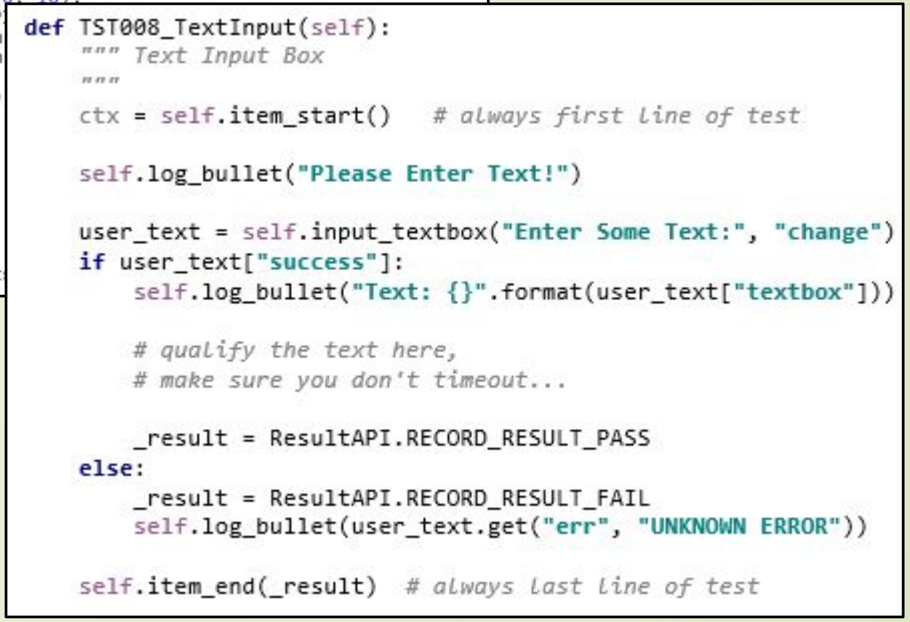
Sistemi Lente/Prism Test Platform

Tests programmed in Python

- Each test item from the JSON script (previous slide), is a python coded function
  - APIs to make test driver code easy
  - Store any measurement
  - Get user input (buttons, text entry)
  - Set dB keys (ex serial number)
  - Add logs
- Vast Python Module Ecosystem to draw upon
  - PyVISA - Test Instrument Control Library
- Online Documentation and Examples



```
def TST000_Meas(self):
    """ Measurement example, with multiple failure messages
    - example of taking multiple measurements, and sending as a list of results
    - if any test fails, this test item fails

        {"id": "TST000_Meas",     "enable": true, "args": {"min": 0, "max": 10},
                                  "fail": [ {"fid": "TST000-0", "msg": "Component apple R1"},
                                            {"fid": "TST000-1", "msg": "Component banana R1"}] },
    :return:
    """
    ctx = self.item_start()   # always first line of test

    time.sleep(self.DEMO_TIME_DELAY * random() * self.DEMO_TIME_RND_ENABLE)

    FAIL_APPLE    = 0  # indexes into the "fail" list, just for code readability
    FAIL_BANANNA = 1

    measurement_results = []  # list for all the coming measurements...

    # Apples measurement...
    _result, _bullet = ctx.record.measurement("apples",
                                              random(),
                                              ResultAPI.UNIT_DB,
                                              ctx.item.args.min,
                                              ctx.item.args.max)
    # if failed, there is a msg in script to attach to the record, for repair purposes
    if _result == ResultAPI.RECORD_RESULT_FAIL:
        msg = ctx.item.fail[FAIL_APPLE]
        ctx.record.fail_msg(msg)

    self.log_bullet(_bullet)
    measurement_results.append(_result)

    # Bananas measurement...
    _result, _bullet = ctx.record.measurement("bananas",
                                              randint(0, 10),
                                              ResultAP
                                              ctx.item
                                              ctx.item

    # if failed, there is a msg in script to attach to
    if _result == ResultAPI.RECORD_RESULT_FAIL:
        msg = ctx.item.fail[FAIL_BANANNA]
        ctx.record.fail_msg(msg)

    self.log_bullet(_bullet)
    measurement_results.append(_result)

    # Note that we can send a list of measurements
    self.item_end(item_result_state=measurement_result
```

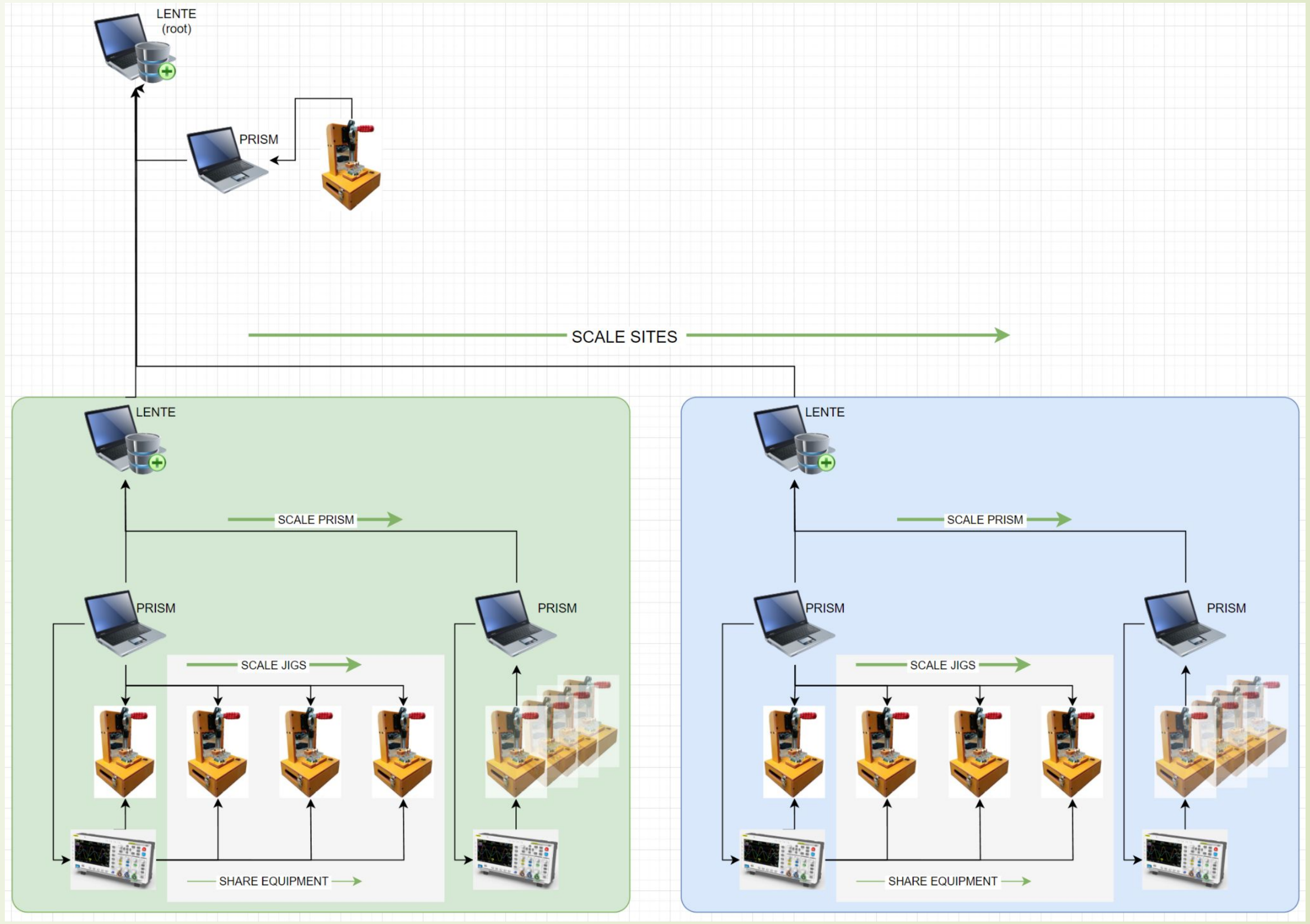


```
def TST008_TextInput(self):
    """ Text Input Box
    """
    ctx = self.item_start()   # always first line of test

    self.log_bullet("Please Enter Text!")

    user_text = self.input_textbox("Enter Some Text:", "change")
    if user_text["success"]:
        self.log_bullet("Text: {}".format(user_text["textbox"]))

        # qualify the text here,
        # make sure you don't timeout...

        _result = ResultAPI.RECORD_RESULT_PASS
    else:
        _result = ResultAPI.RECORD_RESULT_FAIL
        self.log_bullet(user_text.get("err", "UNKNOWN ERROR"))

    self.item_end(_result)  # always last line of test
```

Sistemi Lente/Prism Test Platform

Scaling

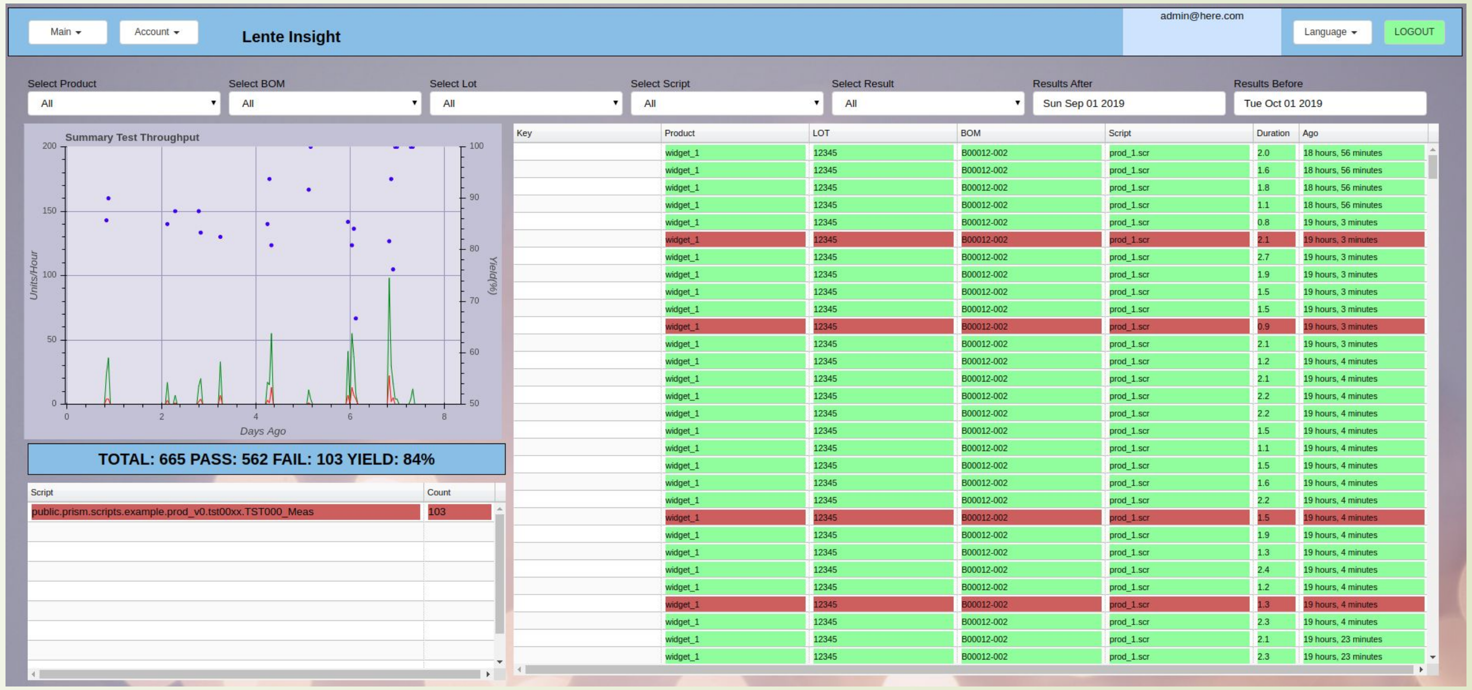
- Prism Computer can support up to 4 Test Jigs
  - Can share equipment
- Each Lente supports a "site"
- Root Lente collects all data
- Pyramid Layout



Sistemi Lente/Prism Test Platform

Lente Production Monitoring Dashboard

- Realtime results
- Can be on or off site (cloud)
- Transfers results into Postgres Database
- Shows Prism Test Station(s) status
- Manage Users and Scripts deployed
- Select Filters to drill down to specific results

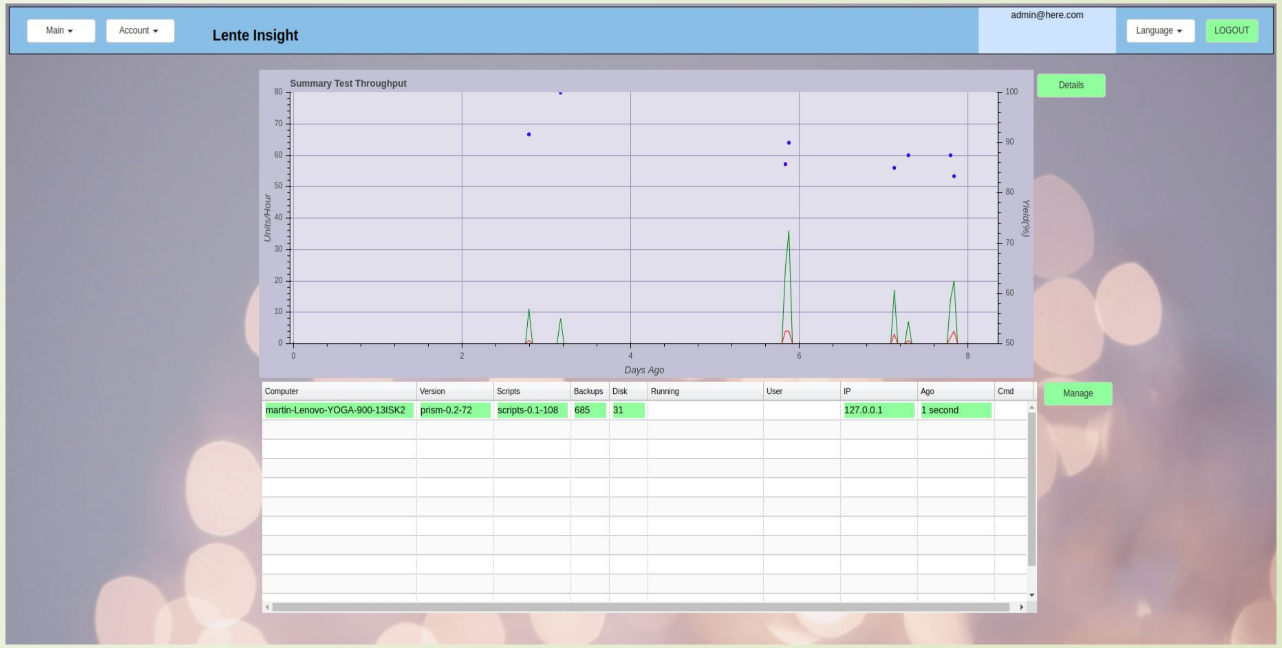


Select Product: All | Select BOM: All | Select Lot: All | Select Script: All | Select Result: All | Results After: Sun Sep 01 2019 | Results Before: Tue Oct 01 2019

TOTAL: 665 PASS: 562 FAIL: 103 YIELD: 84%

Script	Count
public.prism.scripts.example.prod_v0.tst00xx.TST000_Meas	103

Key	Product	LOT	BOM	Script	Duration	Ago
	widget_1	12345	B00012-002	prod_1.scr	2.0	18 hours, 56 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.6	18 hours, 56 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.8	18 hours, 56 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.1	18 hours, 56 minutes
	widget_1	12345	B00012-002	prod_1.scr	0.8	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.1	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.7	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.9	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.5	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.5	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	0.9	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.1	19 hours, 3 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.2	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.1	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.2	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.2	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.5	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.1	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.5	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.6	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.2	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.5	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.9	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.3	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.4	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.2	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	1.3	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.3	19 hours, 4 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.1	19 hours, 23 minutes
	widget_1	12345	B00012-002	prod_1.scr	2.3	19 hours, 23 minutes

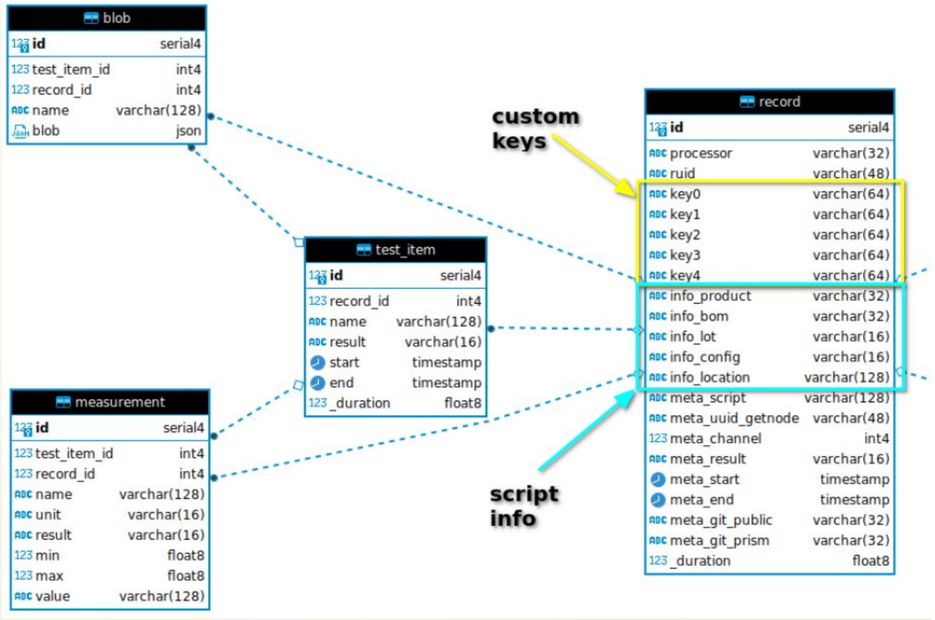


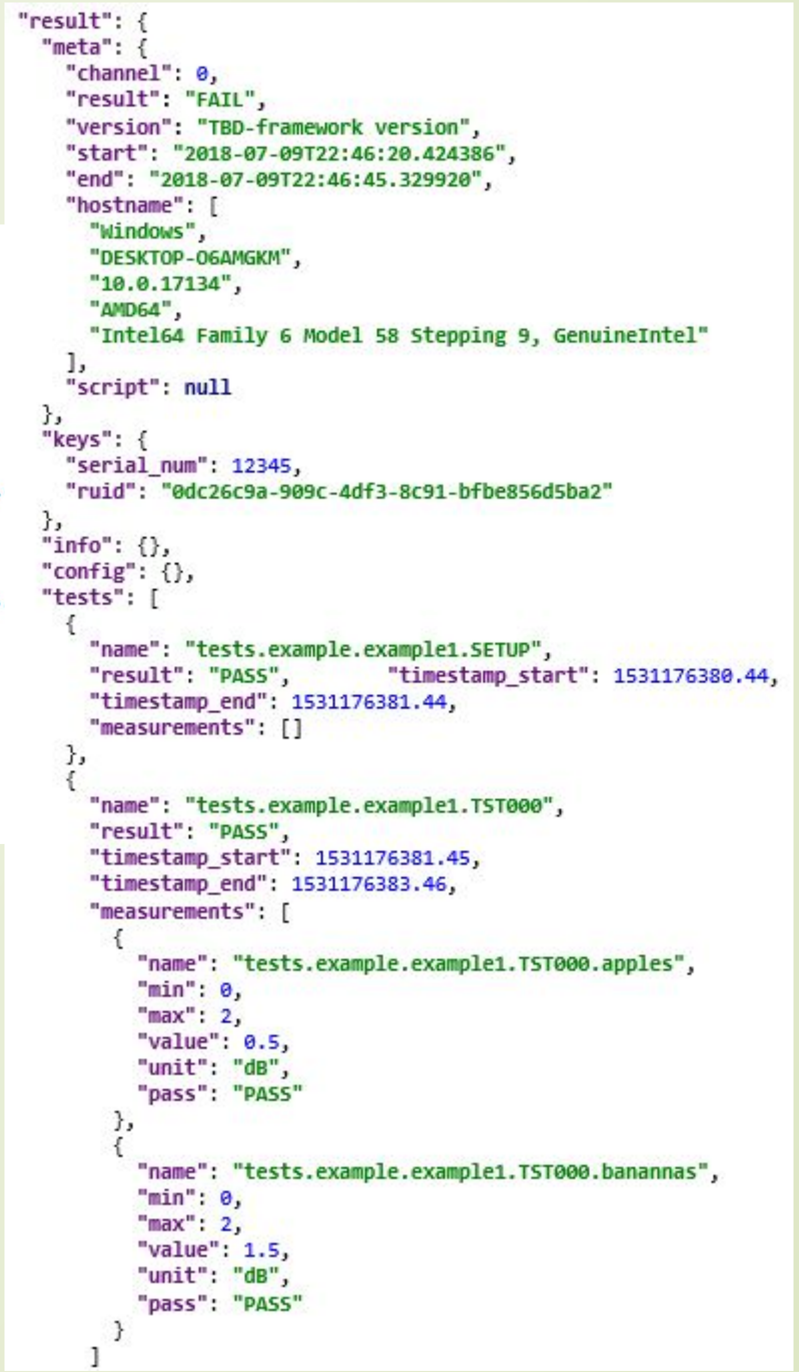
Computer	Version	Scripts	Backups	Disk	Running	User	IP	Ago	Cmd
martin-Lenovo-YOGA-900-13ISK2	prism-0.2-72	scripts-0.1-108	685	31			127.0.0.1	1 second	

Sistemi Lente/Prism Test Platform

Database and JSON Results

- Backend "normalized" SQL Database
- All test results stored in a consistent way to make SQL queries easier
- Postgres
  - Secure, scalable, cloud options
  - JSON BLOB data





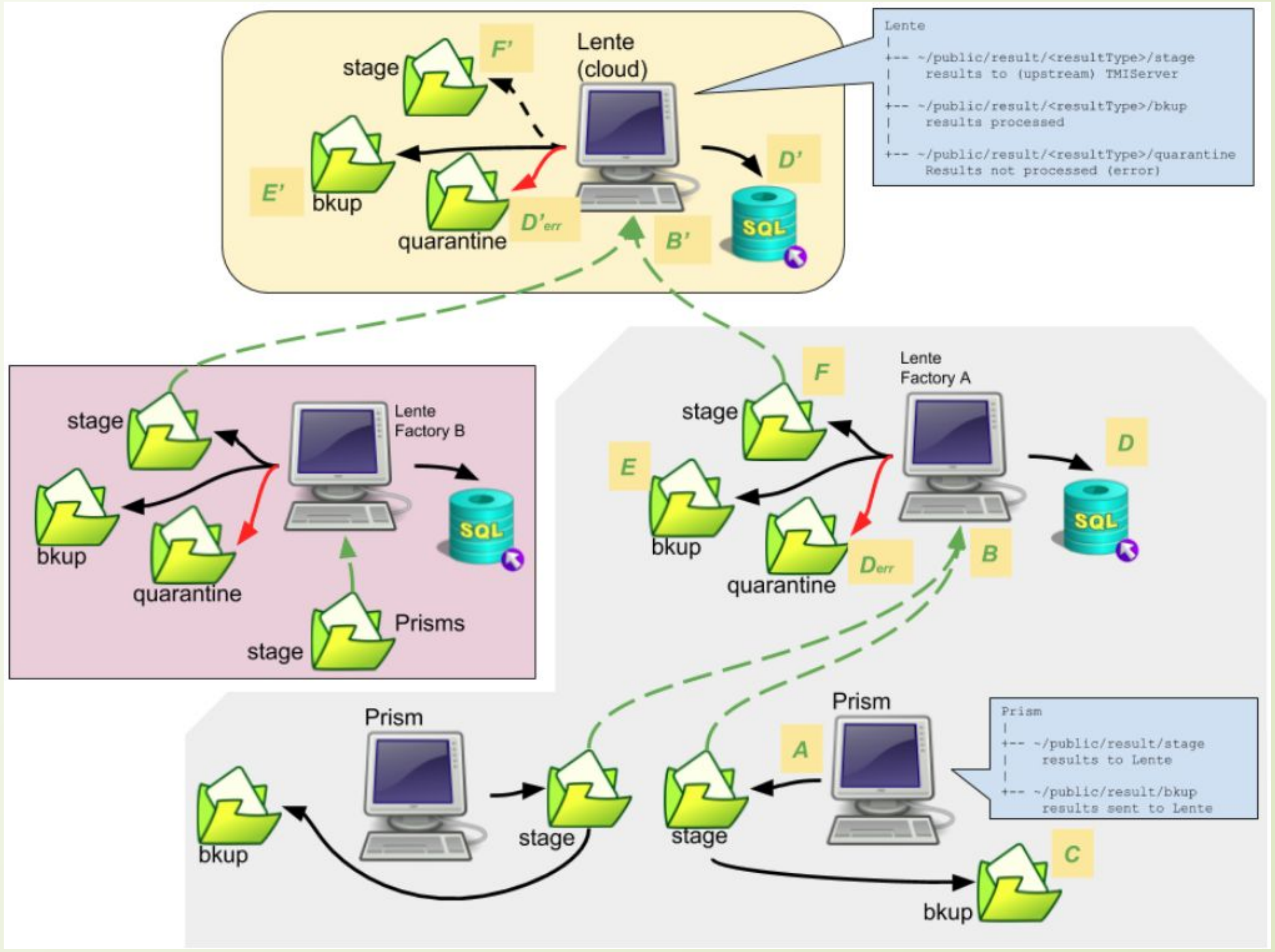
```
"result": {
  "meta": {
    "channel": 0,
    "result": "FAIL",
    "version": "TBD-framework version",
    "start": "2018-07-09T22:46:20.424386",
    "end": "2018-07-09T22:46:45.329920",
    "hostname": [
      "Windows",
      "DESKTOP-O6AMGKM",
      "10.0.17134",
      "AMD64",
      "Intel64 Family 6 Model 58 Stepping 9, GenuineIntel"
    ],
    "script": null
  },
  "keys": {
    "serial_num": 12345,
    "ruid": "0dc26c9a-909c-4df3-8c91-bfbe856d5ba2"
  },
  "info": {},
  "config": {},
  "tests": [
    {
      "name": "tests.example.example1.SETUP",
      "result": "PASS",         "timestamp_start": 1531176380.44,
      "timestamp_end": 1531176381.44,
      "measurements": []
    },
    {
      "name": "tests.example.example1.TST000",
      "result": "PASS",
      "timestamp_start": 1531176381.45,
      "timestamp_end": 1531176383.46,
      "measurements": [
        {
          "name": "tests.example.example1.TST000.apples",
          "min": 0,
          "max": 2,
          "value": 0.5,
          "unit": "dB",
          "pass": "PASS"
        },
        {
          "name": "tests.example.example1.TST000.banannas",
          "min": 0,
          "max": 2,
          "value": 1.5,
          "unit": "dB",
          "pass": "PASS"
        }
      ]
```

PostgreSQL
the world's most advanced open source database

Sistemi Lente/Prism Test Platform

Deployment Architecture

- Multiple Sites
  - scalable
- Pyramid structure
  - Results (optionally) backed up at every level
  - Top of pyramid captures results from all sources
  - Sites don't have access to other sites results
- Each Lente has a local SQL database
  - Local dashboarding
  - Local SQL queries



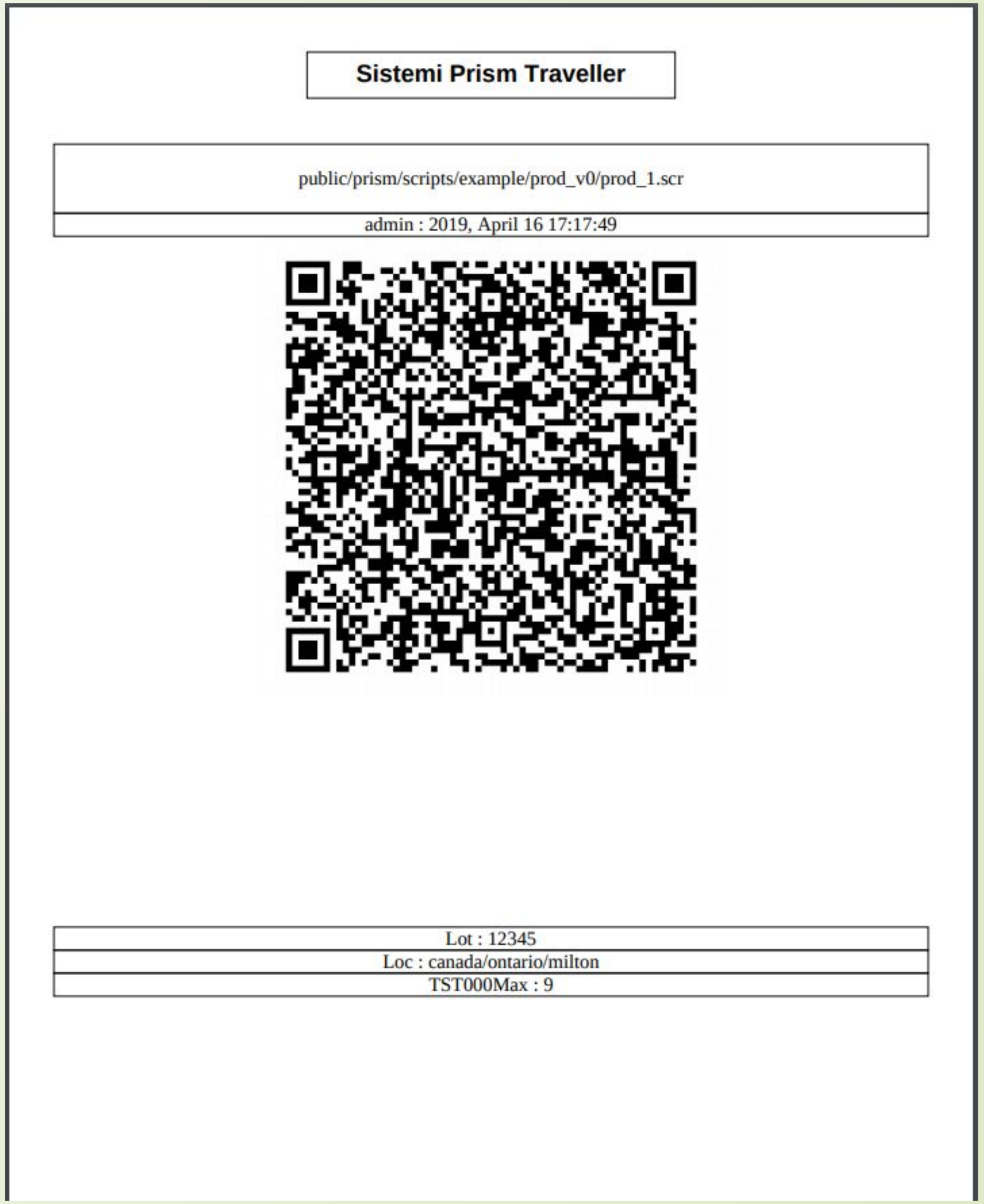
```
Lente
|
+-- ~/public/result/<resultType>/stage
|     results to (upstream) TMIServer
|
+-- ~/public/result/<resultType>/bkup
|     results processed
|
+-- ~/public/result/<resultType>/quarantine
      Results not processed (error)
```

```
Prism
|
+-- ~/public/result/stage
|     results to Lente
|
+-- ~/public/result/bkup
      results sent to Lente
```

Sistemi Lente/Prism Test Platform

Traveler

- Travels with product lot within manufacturing process
- Automates Test Configuration
- No Manual entry
- Scan and Go
- User Defined Production Tracking is encoded into the barcode
- Barcode is encrypted



Sistemi Prism Traveller

public/prism/scripts/example/prod_v0/prod_1.scr

admin : 2019, April 16 17:17:49

Lot : 12345

Loc : canada/ontario/milton

TST000Max : 9

Sistemi Lente/Prism Test Platform

Security

- Stations use Linux file/user security
- Lente/Prism run as Docker containers, and run automatically when PC is booted
- Lente/Prism are hosted in the Google Chrome browser
  - HTTPS supported
- Scripts, Configuration Files, Results, etc, are not accessible by an operator (linux) login account
- Scripts are also additionally protected by an encryption manifest (files can be read, but not changed)
- Results are optionally encrypted
- User Roles allow access to application functions
- Local Prism/Lente Settings file sets passwords

Sistemi Lente/Prism Test Platform

Appendix

Additional Notes

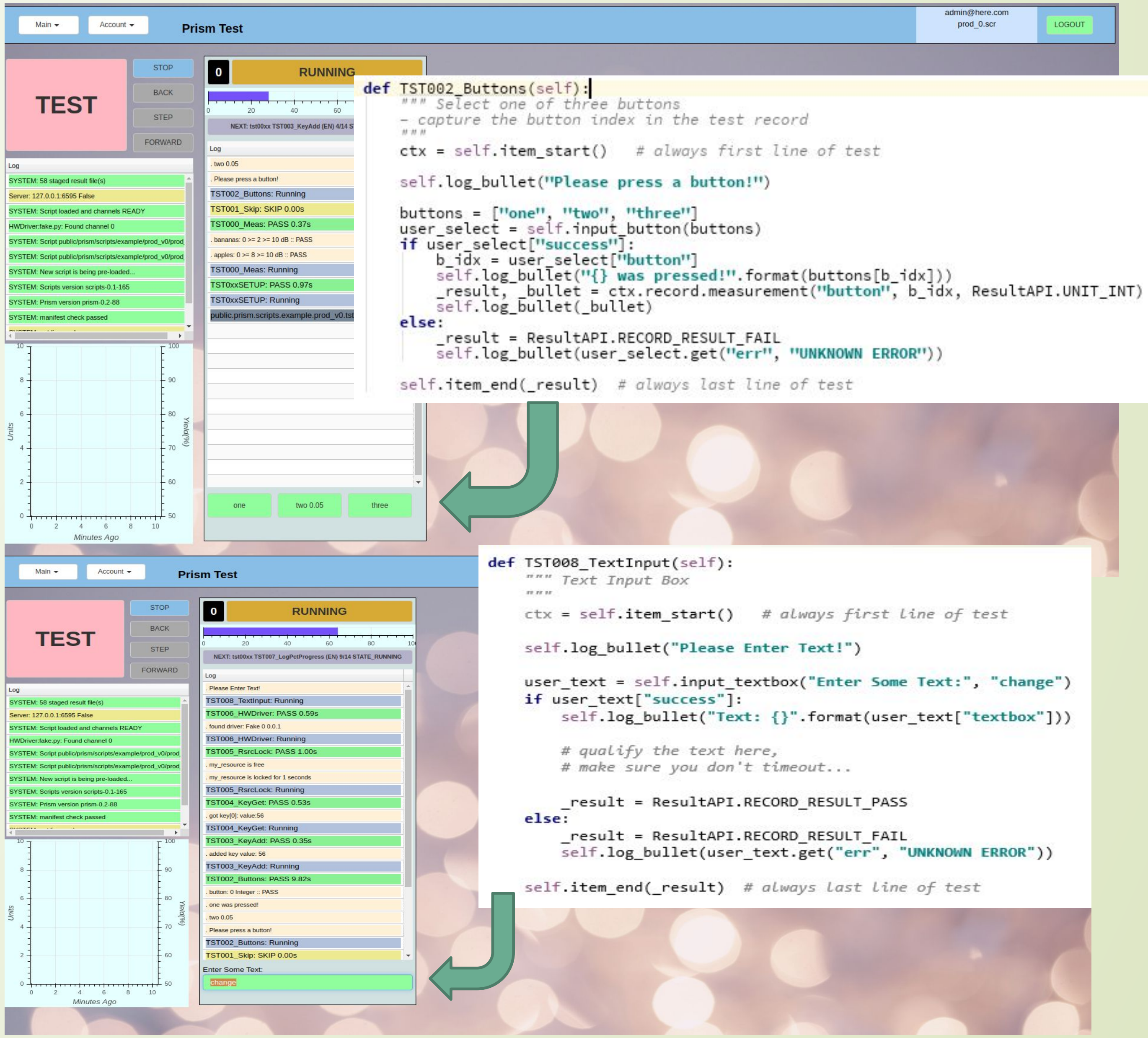
Sistemi Lente/Prism Test Platform

System Requirements

- Prism/Lente PC
  - Laptop preferred (built in UPS)
  - x86
  - “i3” class or better
  - 4+ GB RAM
  - 128+ GB Flash
    - Lente may have larger depending on test volume
  - Ubuntu 22
  - USB port(s)
  - Wired Ethernet (WiFi connections are not recommended in production environments)

Sistemi Prism

- Python 3.10
- Supports up to 4 Test Jigs per PC
  - All Jigs can run the same test script asynchronously
- Allows resource sharing between Jigs
  - For example, one expensive test equipment can be shared between jigs reducing cost
- User defined Buttons
- Text Entry (scanners)



```
NEXT: tst00xx TST003_KeyAdd (EN) 4/14 S
```

```
Log
SYSTEM: 58 staged result file(s)
Server: 127.0.0.1:6595 False
SYSTEM: Script loaded and channels READY
HWDriver:fake.py: Found channel 0
SYSTEM: Script public/prism/scripts/example/prod_v0/prod_
SYSTEM: Script public/prism/scripts/example/prod_v0/prod_
SYSTEM: New script is being pre-loaded...
SYSTEM: Scripts version scripts-0.1-165
SYSTEM: Prism version prism-0.2-88
SYSTEM: manifest check passed
```

```
Log
. two 0.05
. Please press a button!
TST002_Buttons: Running
TST001_Skip: SKIP 0.00s
TST000_Meas: PASS 0.37s
. bananas: 0 >= 2 >= 10 dB :: PASS
. apples: 0 >= 8 >= 10 dB :: PASS
TST000_Meas: Running
TST0xxSETUP: PASS 0.97s
TST0xxSETUP: Running
public.prism.scripts.example.prod_v0.tst
```

```python
def TST002_Buttons(self):
    """ Select one of three buttons
    - capture the button index in the test record
    """
    ctx = self.item_start()   # always first line of test

    self.log_bullet("Please press a button!")

    buttons = ["one", "two", "three"]
    user_select = self.input_button(buttons)
    if user_select["success"]:
        b_idx = user_select["button"]
        self.log_bullet("{} was pressed!".format(buttons[b_idx]))
        _result, _bullet = ctx.record.measurement("button", b_idx, ResultAPI.UNIT_INT)
        self.log_bullet(_bullet)
    else:
        _result = ResultAPI.RECORD_RESULT_FAIL
        self.log_bullet(user_select.get("err", "UNKNOWN ERROR"))

    self.item_end(_result)  # always last line of test
```

```
NEXT: tst00xx TST007_LogPctProgress (EN) 9/14 STATE_RUNNING
```

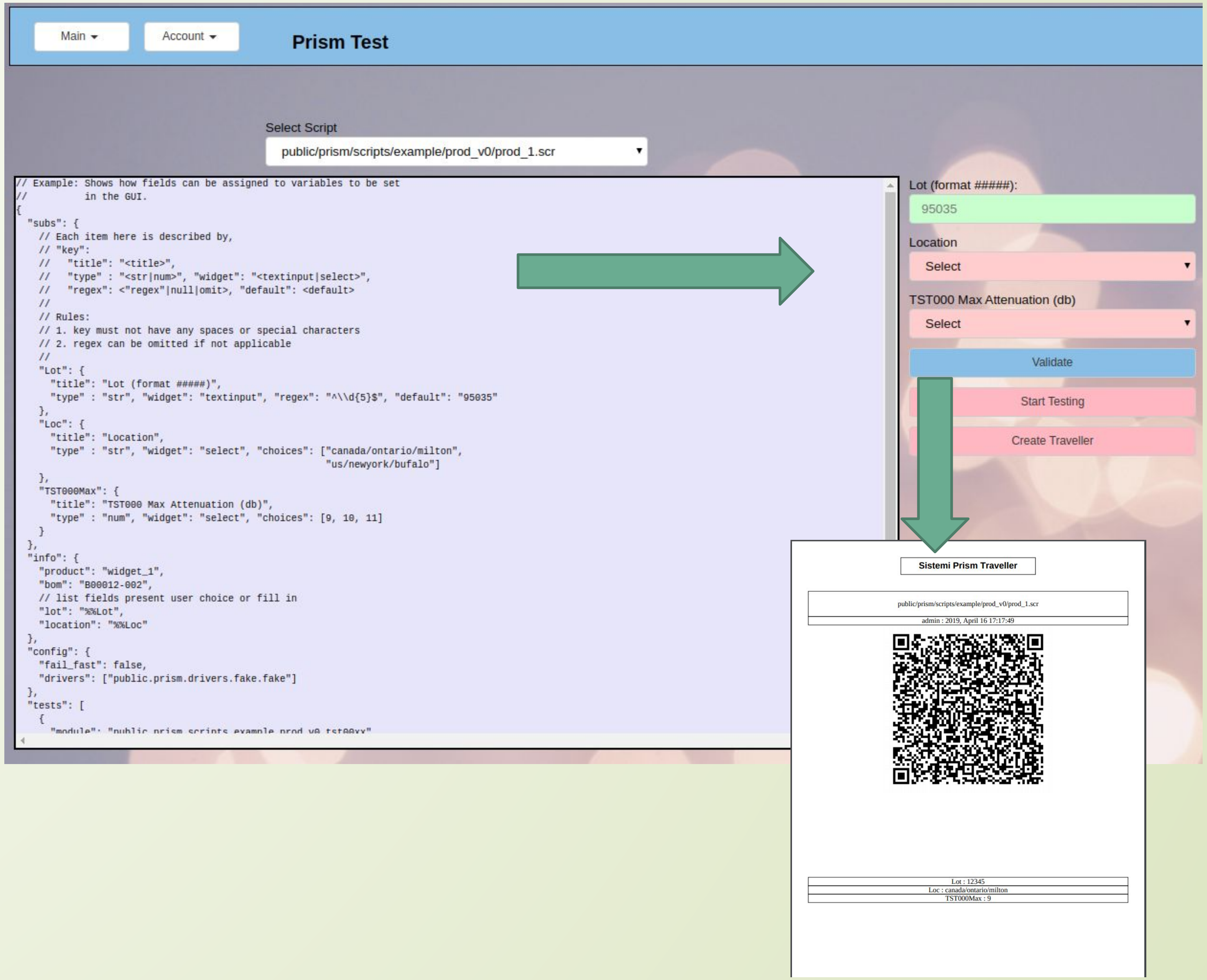
```
Log
SYSTEM: 58 staged result file(s)
Server: 127.0.0.1:6595 False
SYSTEM: Script loaded and channels READY
HWDriver:fake.py: Found channel 0
SYSTEM: Script public/prism/scripts/example/prod_v0/prod_
SYSTEM: Script public/prism/scripts/example/prod_v0/prod_
SYSTEM: New script is being pre-loaded...
SYSTEM: Scripts version scripts-0.1-165
SYSTEM: Prism version prism-0.2-88
SYSTEM: manifest check passed
```

```
Log
. Please Enter Text!
TST008_TextInput: Running
TST006_HWDriver: PASS 0.59s
. found driver: Fake 0 0.0.1
TST006_HWDriver: Running
TST005_RsrcLock: PASS 1.00s
. my_resource is free
. my_resource is locked for 1 seconds
TST005_RsrcLock: Running
TST004_KeyGet: PASS 0.53s
. got key[0]: value:56
TST004_KeyGet: Running
TST003_KeyAdd: PASS 0.35s
. added key value: 56
TST003_KeyAdd: Running
TST002_Buttons: PASS 9.82s
. button: 0 Integer :: PASS
. one was pressed!
. two 0.05
. Please press a button!
TST002_Buttons: Running
TST001_Skip: SKIP 0.00s
Enter Some Text:
change
```

```python
def TST008_TextInput(self):
    """ Text Input Box
    """
    ctx = self.item_start()   # always first line of test

    self.log_bullet("Please Enter Text!")

    user_text = self.input_textbox("Enter Some Text:", "change")
    if user_text["success"]:
        self.log_bullet("Text: {}".format(user_text["textbox"]))

        # qualify the text here,
        # make sure you don't timeout...

        _result = ResultAPI.RECORD_RESULT_PASS
    else:
        _result = ResultAPI.RECORD_RESULT_FAIL
        self.log_bullet(user_text.get("err", "UNKNOWN ERROR"))

    self.item_end(_result)  # always last line of test
```

Sistemi Prism

- JSON Script with GUI variable substitution
  - Drop Down Selection
  - Text Entry validated by Regex
- For example,
  - Lot Number
  - Location
  - Measurement limits
- Traveler can be created from User input(s) for hands free Production floor configuration



Select Script: public/prism/scripts/example/prod_v0/prod_1.scr

```
// Example: Shows how fields can be assigned to variables to be set
//          in the GUI.
{
  "subs": {
    // Each item here is described by,
    // "key":
    //   "title": "<title>",
    //   "type" : "<str|num>", "widget": "<textinput|select>",
    //   "regex": <"regex"|null|omit>, "default": <default>
    //
    // Rules:
    // 1. key must not have any spaces or special characters
    // 2. regex can be omitted if not applicable
    //
    "Lot": {
      "title": "Lot (format #####)",
      "type" : "str", "widget": "textinput", "regex": "^\\d{5}$", "default": "95035"
    },
    "Loc": {
      "title": "Location",
      "type" : "str", "widget": "select", "choices": ["canada/ontario/milton",
                                                      "us/newyork/bufalo"]
    },
    "TST000Max": {
      "title": "TST000 Max Attenuation (db)",
      "type" : "num", "widget": "select", "choices": [9, 10, 11]
    }
  },
  "info": {
    "product": "widget_1",
    "bom": "B00012-002",
    // list fields present user choice or fill in
    "lot": "%%Lot",
    "location": "%%Loc"
  },
  "config": {
    "fail_fast": false,
    "drivers": ["public.prism.drivers.fake.fake"]
  },
  "tests": [
    {
      "module": "public prism scripts example prod v0 tst00xx"
```

Lot (format #####): 95035

Location: Select

TST000 Max Attenuation (db): Select

Sistemi Prism Traveller

public/prism/scripts/example/prod_v0/prod_1.scr

admin : 2019, April 16 17:17:49

Lot : 12345

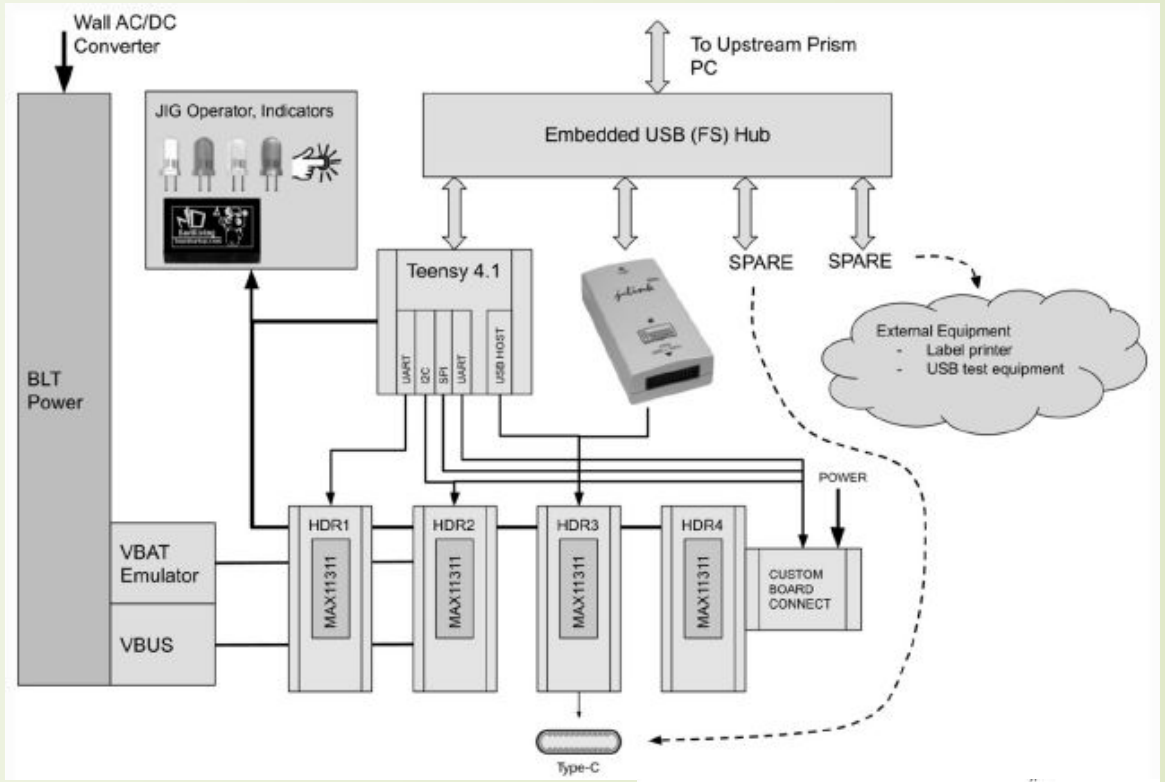
Loc : canada/ontario/milton

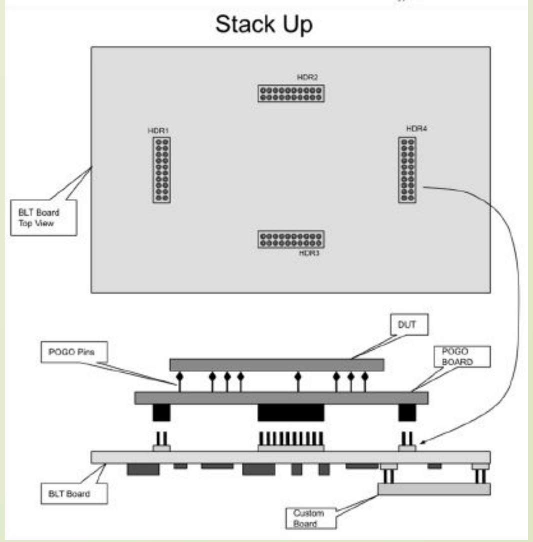
TST000Max : 9

Sistemi Lente/Prism Test Platform

Bed Of Nails Design (BOND)

- Multipurpose Testing board
  - Open Source Hardware
- Teensy 4.1 Controller
  - 600MHz, 512KRAM, 8MB Flash, SD card, etc
  - Prism Arduino RPC Server
- Supplies
  - VBUS (5V, 1 Amp)
  - VBAT - two quadrant battery emulator, 0.5-5V, 2 Amp
  - Programmable LDO
- MAX11311 IO Chip (X4)
  - 12 port ADC, DAC, GPIO
- On board USB HUB (reduces wiring)
- Segger JTAG
- Jig UI (LEDs, Buttons)
- more…
- Prism "driver"
  - Python APIs to access BOND





- Separate Pogo board for DUT test pad layout
- Expansion Hat