Python-Based Application Acceleration Through Computational Storage Drive

Presenter(s):

Raj Kumar Dani

Associate Director,

Samsung Semiconductor Inc

Jaysankar O P

Associate Technical Director,

Samsung Semiconductor Inc

Arun V Pillai

Staff Engineer,

Samsung Semiconductor Inc

Disclaimer

This presentation and/or accompanying oral statements by Samsung representatives collectively, the "Presentation" is intended to provide information concerning the SSD and memory industry and Samsung Electronics Co., Ltd. and certain affiliates (collectively, "Samsung").While Samsung strives to provide information that is accurate and up-to-date, this Presentation may nonetheless contain inaccuracies or omissions. As a consequence, Samsung does not in any way guarantee the accuracy or completeness of the information provided in this Presentation.

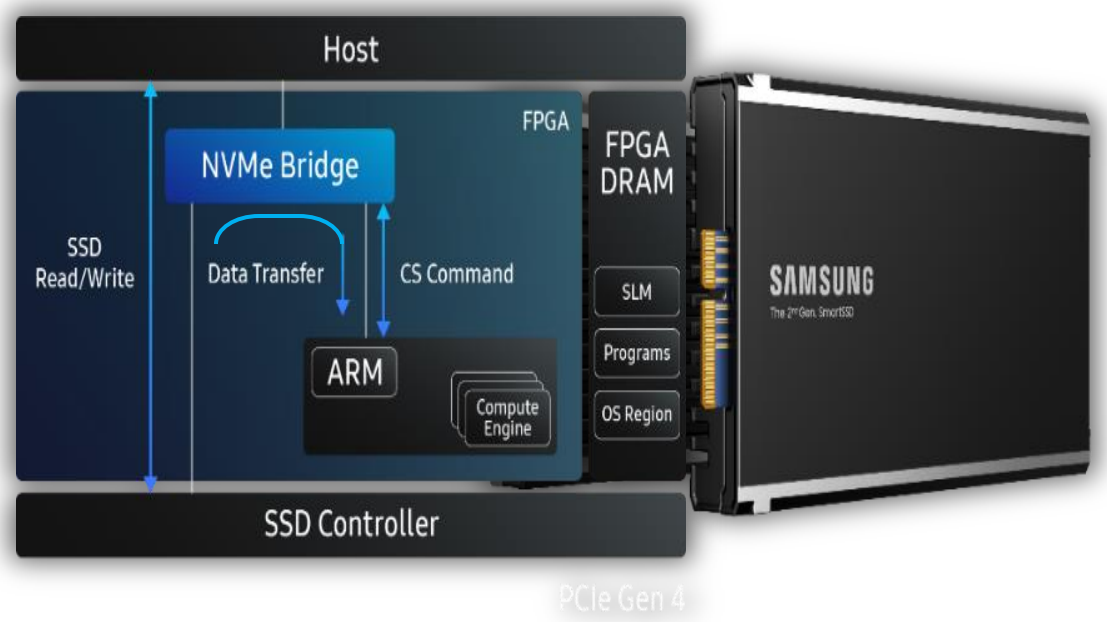
This Presentation may include forward-looking statements, including, but not limited to, statements about any matter that is not a historical fact; statements regarding Samsung's intentions, beliefs or current expectations concerning, among other things, market prospects, technological developments, growth, strategies, and the industry in which Samsung operates; and statements regarding products or features that are still in development. By their nature, forward-looking statements involve risks and uncertainties, because they relate to events and depend on circumstances that may or may not occur in the future. Samsung cautions you that forward looking statements are not guarantees of future performance and that the actual developments of Samsung, the market, or industry in which Samsung operates may differ materially from those made or suggested by the forward-looking statements in this Presentation. In addition, even if such forward-looking statements are shown to be accurate, those developments may not be indicative of developments in future periods.

Agenda

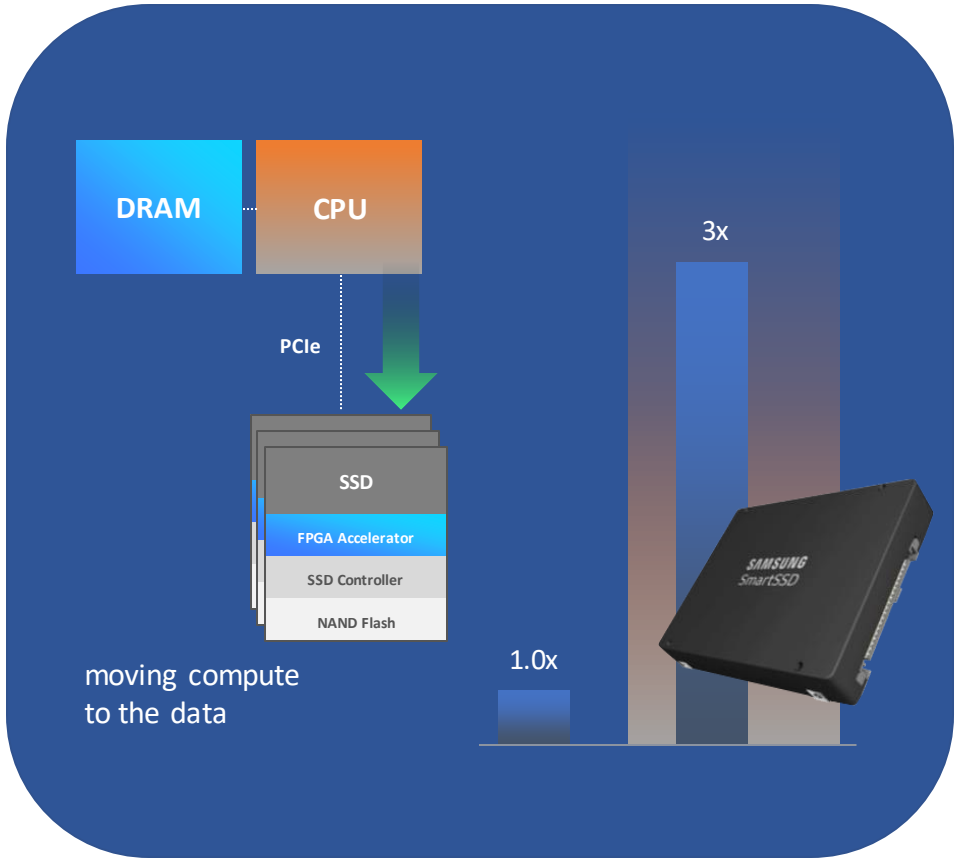
- Samsung SmartSSD®
- Software Stack for Computational Storage Drive (CSD)
- Python Interface to CSD
- CS Python Library plugin
- CSD Test Coverage
- Python Based CSD Monitoring
- CSD Common Use Cases
- SmartSSD®: Geo Locator Using PostgreSQL
- What's Next ..?

Samsung SmartSSD®

- Samsung 2nd Gen SmartSSD based on NVMe standards in development (TP4091)

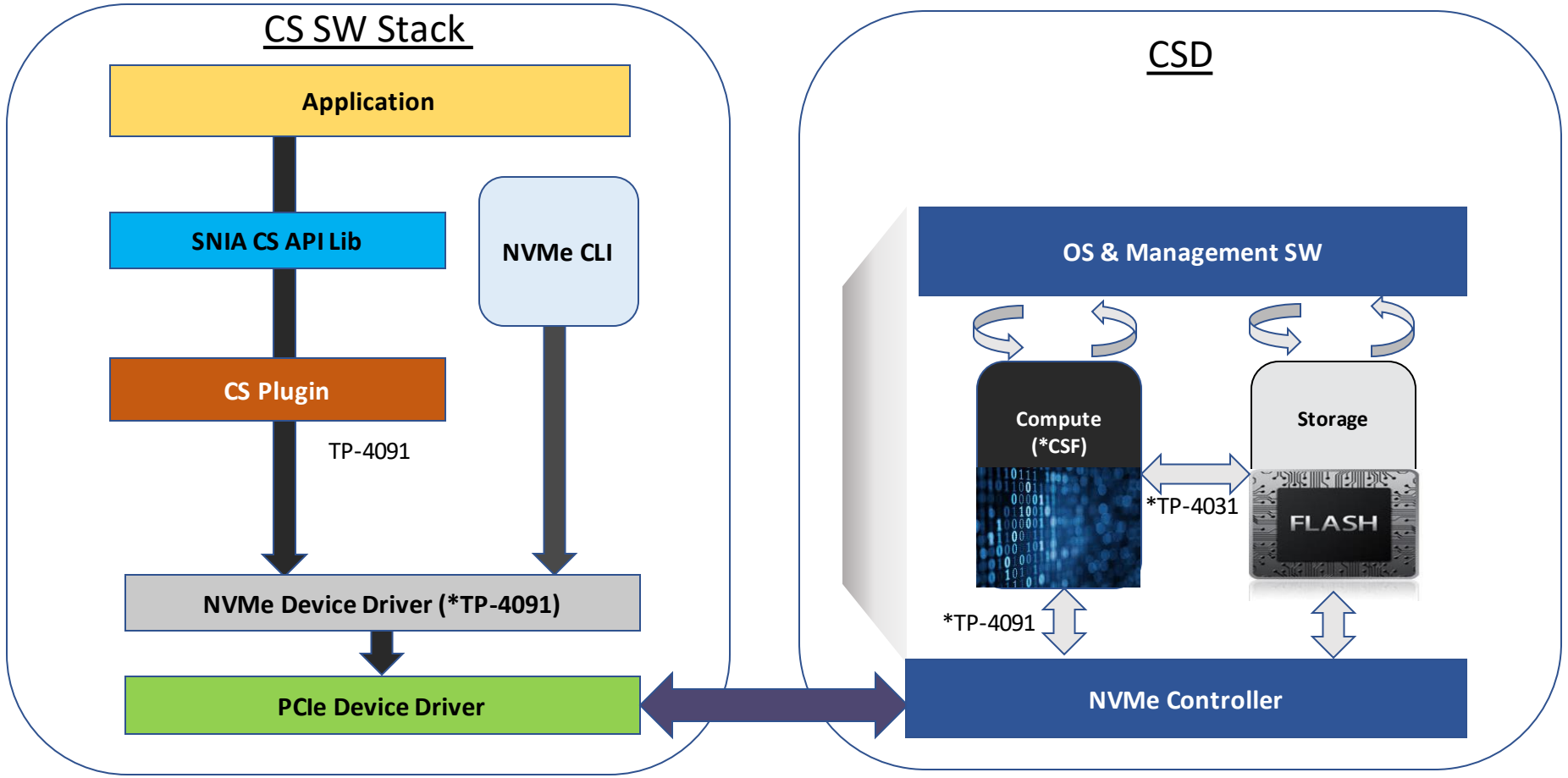


* SLM: Subsystem Local Memory



Software Stack for Computation Storage Drive (CSD)

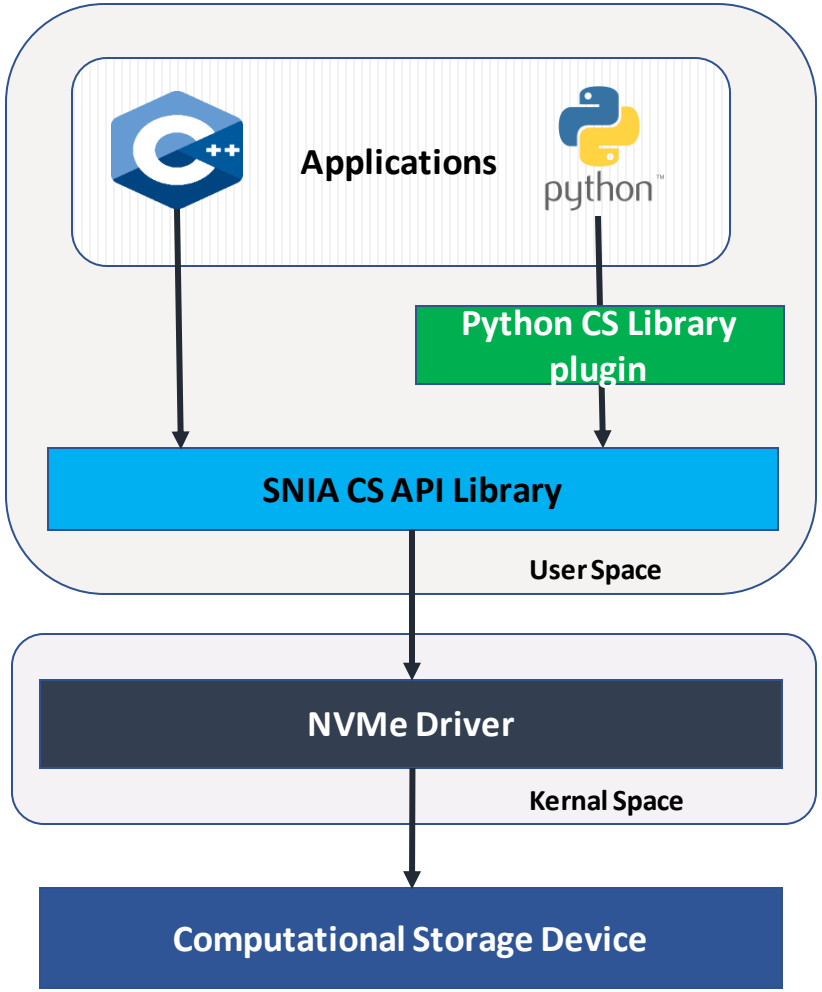
- Computational Storage Drive (CSD) is capable of offloading Host processing to achieve near-storage Compute

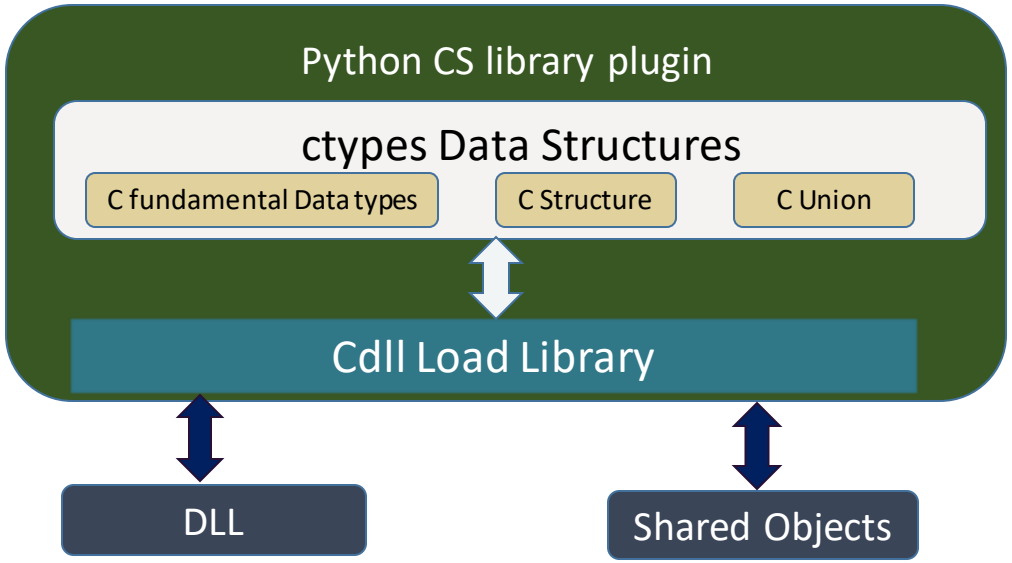


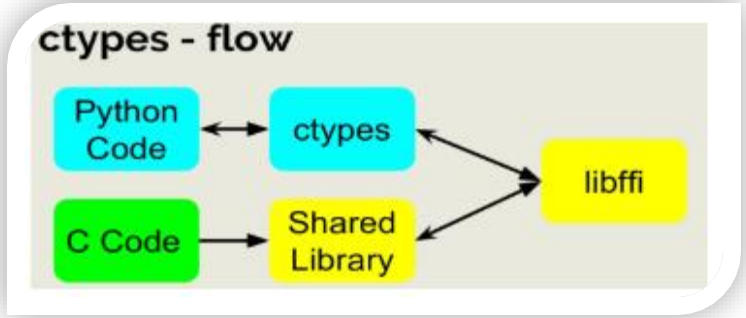
* TP-4091,4031 : NVMe Technical proposals
* CSF : Computational Storage Function

Python Interface with CSD

- Python CS Library is realized using Ctypes module

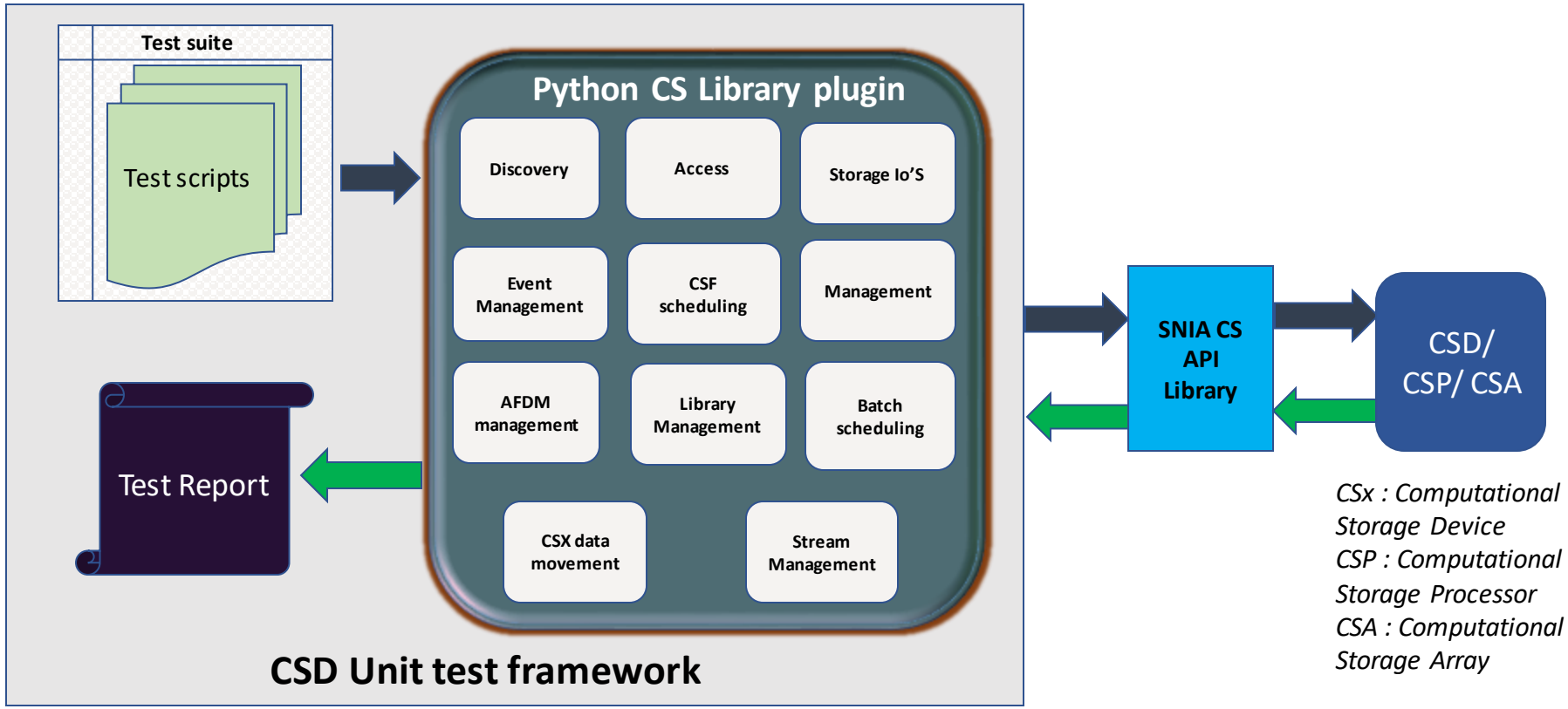






Python CS Library Plugin

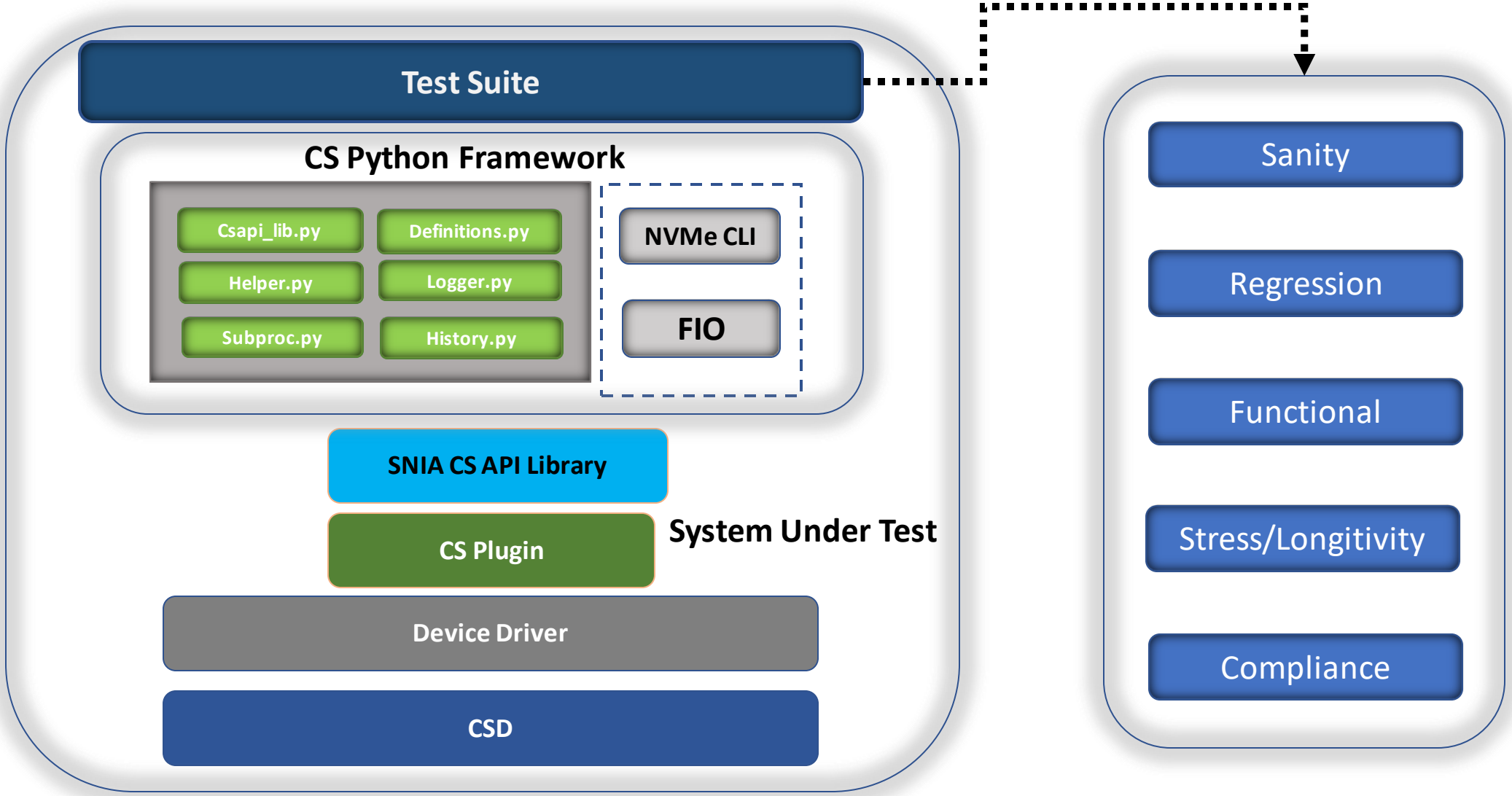
- Validate/verify CSx function through SNIA CS APIs
- Implemented using pytest Framework



CSx : Computational Storage Device
CSP : Computational Storage Processor
CSA : Computational Storage Array

CSD Test Coverage

- E2E automated CSD Test Coverage



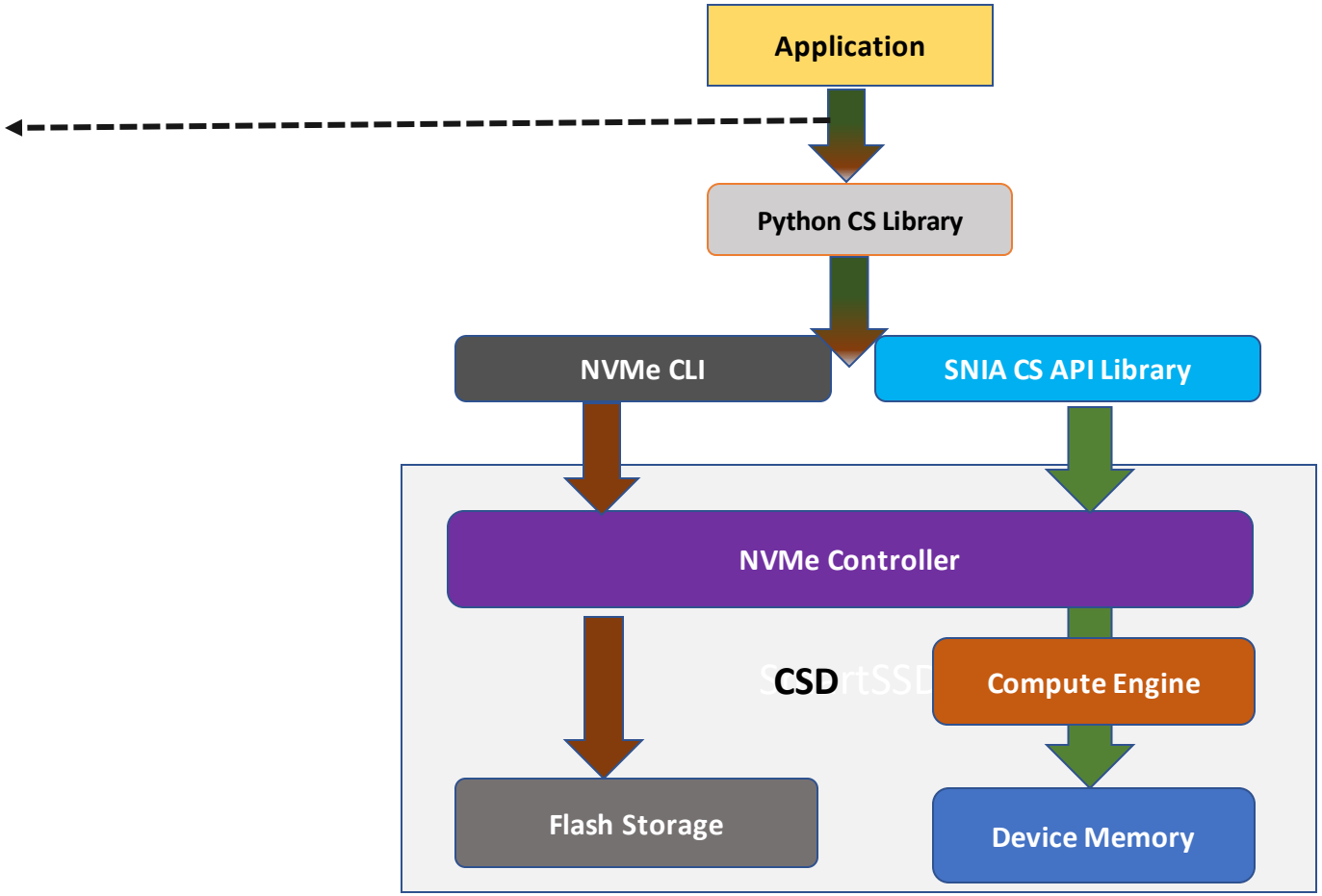
Python Based CSD Monitoring

- JSON based real-time statistics for CS and NVMe Controller
- Easily integrable with applications/use cases

JSON Based Monitoring

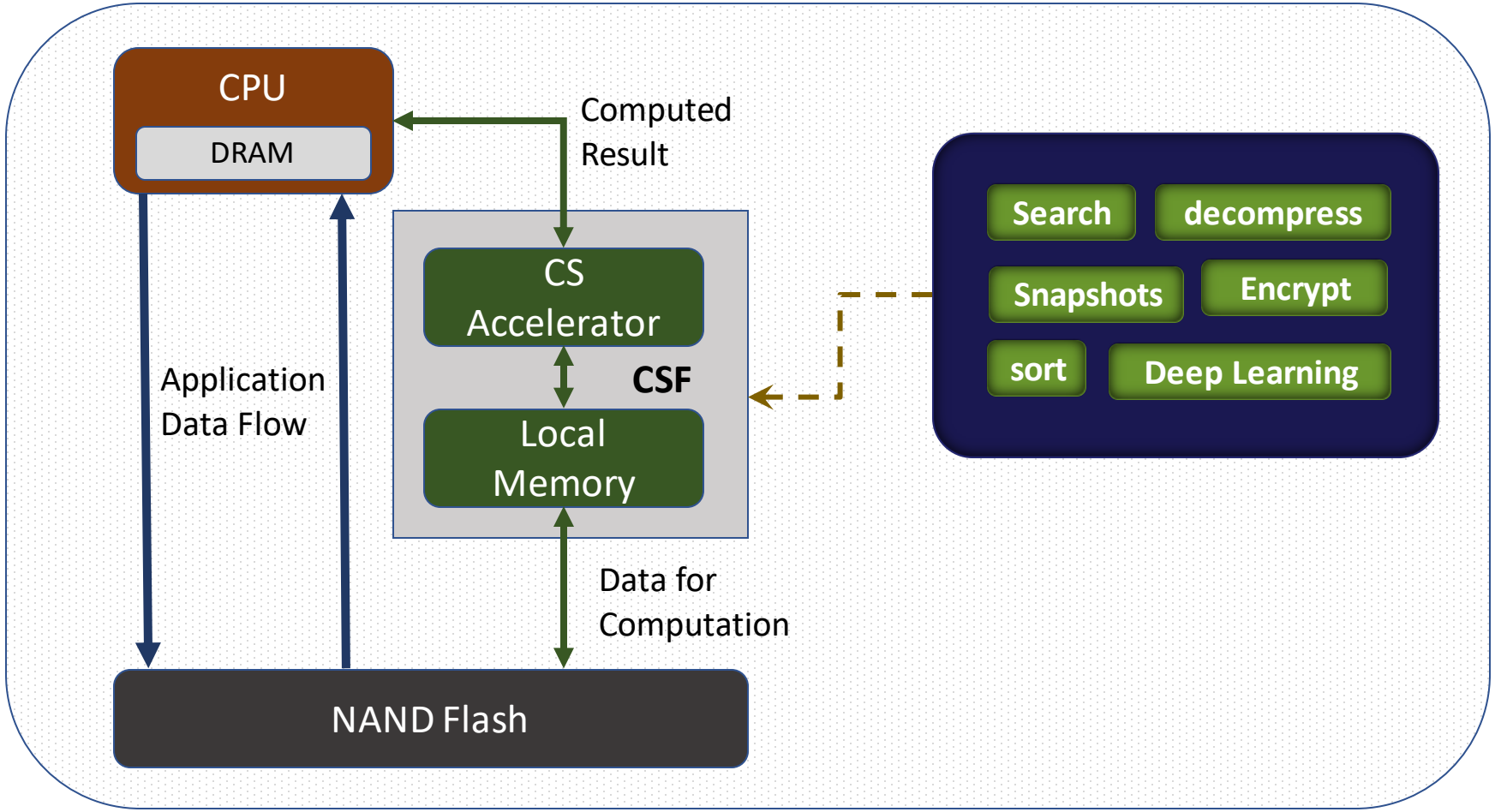
```
{
  "CSFUsage" :
    {
        "TotalUptimeSeconds" :"",
        "TotalExecutions" : "",
        "ShortestTimeUsecs" : "",
        "LongestTimeUsecs":""
    },
 "CsStatsInfo" :
    {
        "MemoryDetails" :""
    }
}
```

Sample output



Common CSD Use cases ..

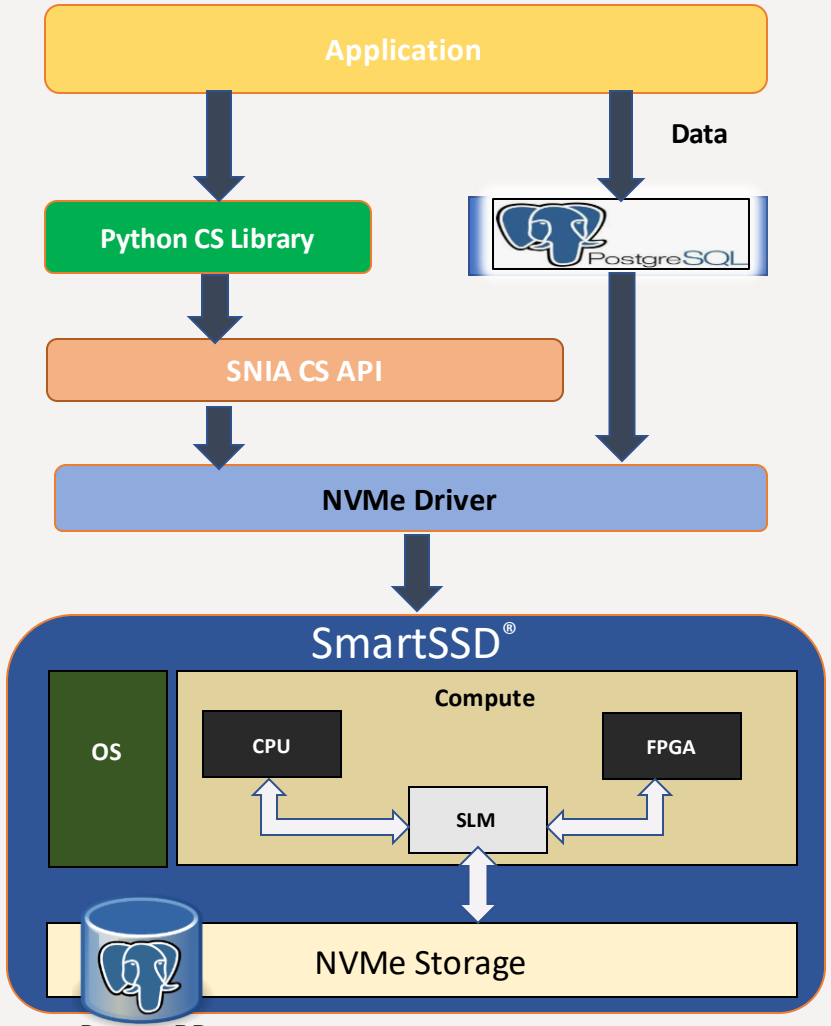
- Near storage compute operations are suitable to offload to computational storage



SmartSSD®: Geo Locator Using PostgreSQL

Flash Memory Summit

- PostGIS and spatial database extender enabled in PostgreSQL for geo location features



id	name	latitude	longitude	country	population	unique_id
48504	Dhaka	23.7289	90.3944	Bangladesh	16839000	1050529279
48505	Buenos Aires	-34.5997	-58.3819	Argentina	16216000	1032717330
48506	Osaka	34.752	135.4582	Japan	15490000	1392419823
48507	Lagos	6.45	3.4	Nigeria	15487000	1566593751
48508	Istanbul	41.01	28.9603	Turkey	15311000	1792756324

(41694 rows)

```
postgres=# select name, country, population, latitude, longitude
postgres-# from citydetails
postgres-# where ((latitude between  -5 and 5)) and population > 2000000;
     name      |     country      | population | latitude | longitude
---------------+------------------+------------+----------+-----------
 Kinshasa      | Congo (Kinshasa) |   15056000 |  -4.3317 |   15.3139
 Kuala Lumpur  | Malaysia         |    8639000 |   3.1478 |  101.6953
 Bogota        | Colombia         |    7743955 |   4.6126 |  -74.0705
 Nairobi       | Kenya            |    5545000 |  -1.2864 |   36.8172
 Singapore     | Singapore        |    5271000 |      1.3 |     103.8
 Timbio        | Colombia         |    4444444 |   2.3528 |  -76.6819
 Guayaquil     | Ecuador          |    2723665 |    -2.19 |  -79.8875
 Cali          | Colombia         |    2471474 |     3.44 |  -76.5197
 Fortaleza     | Brazil           |    2452185 |  -3.7275 |  -38.5275
 Yaounde       | Cameroon         |    2440462 |   3.8578 |   11.5181
 Douala        | Cameroon         |    2446945 |     4.05 |     9.7
 Mogadishu     | Somalia          |    2120000 |   2.0408 |   45.3425
 Medan         | Indonesia        |    2109330 |   3.6667 |   98.6667
 Quito         | Ecuador          |    2011388 |    -0.22 |  -78.5125
(14 rows)
```

Find cities near earth equator having population more than 2000000

```
postgres=# select name, country,
postgres-# (
postgres(#   point(-118.4068, 34.1139)<@>point(local.longitude, local.latitude)
postgres(# )  * 1609.34::int as point_point_distance_in_meters
postgres-# from citydetails,
postgres-# lateral (
postgres(# select id,latitude, longitude from citydetails
postgres(# where ((latitude between  33.1139 and 35.1139)
postgres(# and (longitude between -119.4068 and -118.84297))
postgres(# ) as local
postgres-# where citydetails.id = local.id;
       name       |    country    | point_point_distance_in_meters
------------------+---------------+--------------------------------
 Oxnard           | United States |              71855.61647291823
 Thousand Oaks    | United States |              43972.17997046268
 San Buenaventura | United States |              77888.34782426493
 Santa Paula      | United States |              66442.11945303493
 Camarillo        | United States |              58789.97434961202
 Moorpark         | United States |             47255.503117375905
 Port Hueneme     | United States |              73509.76226287287
 Fillmore         | United States |              56666.34332526211
 Ojai             | United States |              85676.85000993097
 Mira Monte       | United States |              87954.80720469833
 El Rio           | United States |              70504.56735180593
(11 rows)
```

Calculate point-point distance between Los Angeles and near cities in meters

Sample data source : https://simplemaps.com/data/world-cities

Summary

Python CS API Library Enables

- E2E qualification of Computational storage devices
- Provides CSD real time metrics
- Enables applications to be integrated with CSDs easily

What's Next ?..

- More use-case analysis on CSD using python
- 100% test coverage for CSx

THANK YOU