presented by

CLOUDFLARE

Web-Scale UEFI Configuration Management for Heterogeneous Infrastructure

UEFI 2025 Developers Conference & Plugfest

October 9

Nnamdi Ajah

Meet the Presenter

Nnamdi Ajah

Web-Scale UEFI Configuration Management for Heterogeneous Infrastructure

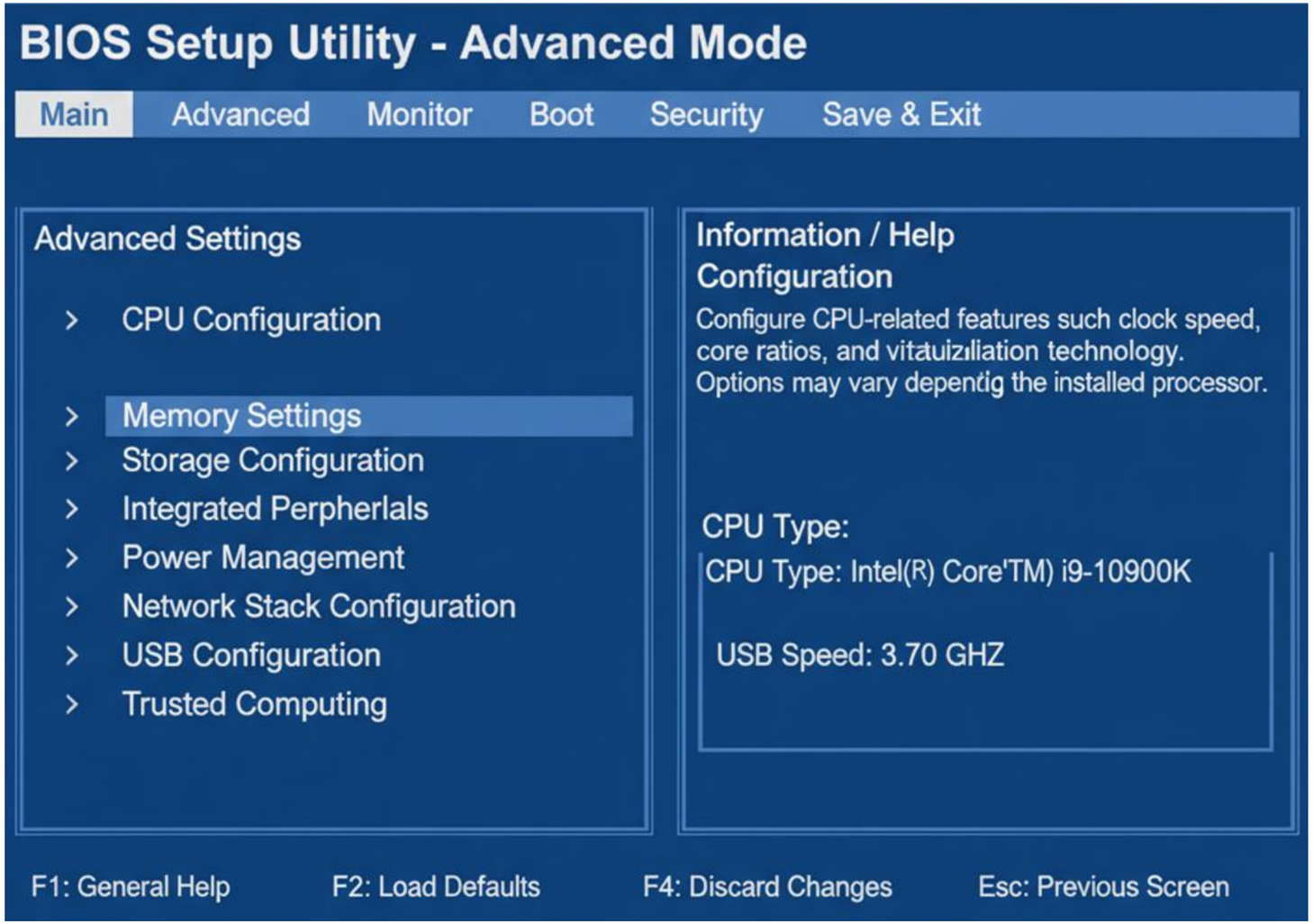
Hardware Systems Engineer, Cloudflare

- OpenBMC developer at Cloudflare.
- Contributor to the Project Argus DC-SCM 2.
- Downhill cycler

Agenda

- Goal
- Challenges
- Solutions Considered
- Vendor-Neutral Design
- Workflow
- Examples

We've All Been Here



Goal

Enable teams to reliably manage UEFI settings without adding complexity or risk.

Key Takeaways

- Vendor-neutral UEFI configuration tool
- Handles complex configurations
- Integrates with existing infrastructure configuration management tools e.g. SaltStack
- Intuitive to use
- Reduces human errors
- Eliminates vendor delays
- Ease of maintenance

Challenges

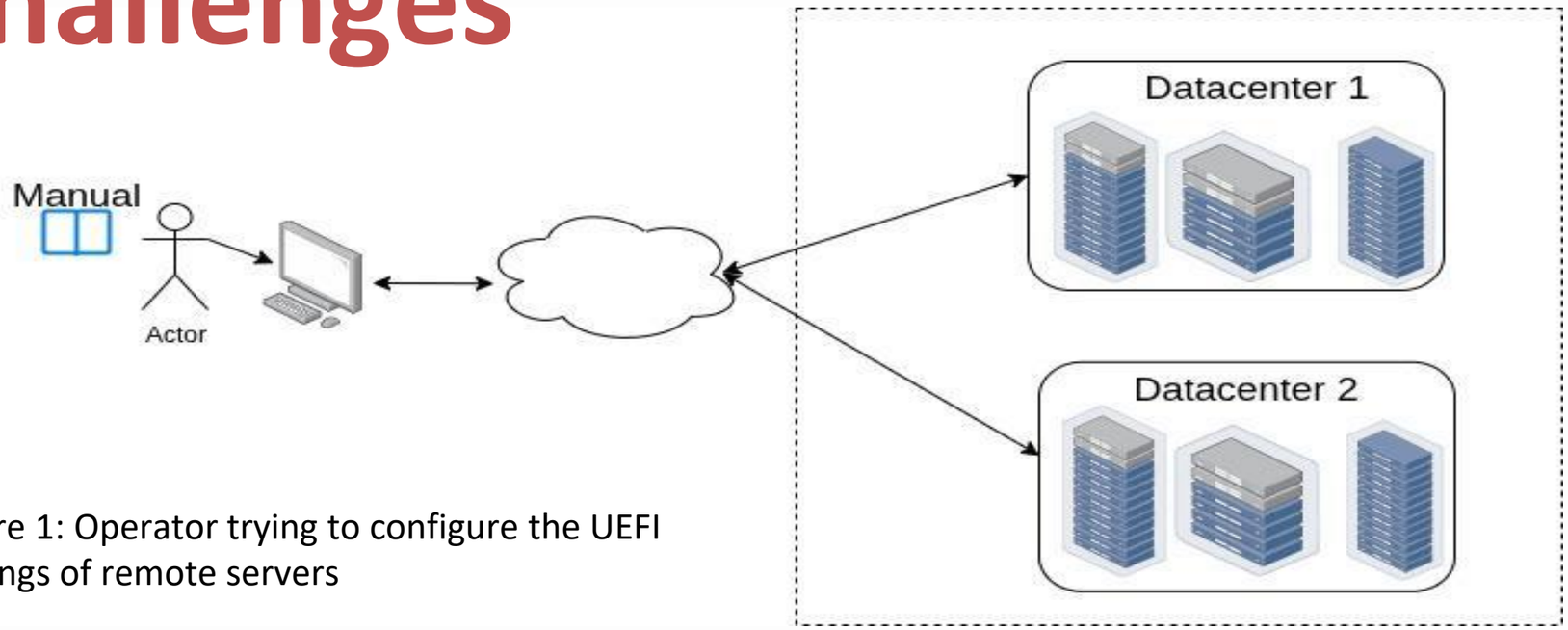


Figure 1: Operator trying to configure the UEFI settings of remote servers

Inexhaustive list of challenges

- Multiple generations of servers from different OEMs globally distributed
- Inconsistent interfaces and documentations from various OEMs
- Varying tooling from different vendors
- Inconsistent redfish support
- Operator may not be technically savvy

Solutions Considered

Solution	Comments
Redfish wrapper	• Complicated layouts • Wrapper can quickly become complex • Won't support non-Human Interface Infrastructure automation use-cases • Redfish must be available at all times
Unifying vendor tooling	• Inconsistent performance e.g. some could take 30 seconds to read a Human Interface Infrastructure config. • Turn-around time interfacing with vendors • Legal issues e.g. Non-Disclosure Agreements
Standalone EFI Config app	• Can work with existing configurations tools e.g. Salt • Eliminates need for vendor-specific knowledge • Work across hardware generations
iPXE patching	• Could not retrieve some settings • Tightly coupled to Preboot Execution Environment (iPXE) • Not easy to use

Vendor Neutral Tool Design

Overview

- Parse prompt strings from UEFI Human Interface Infrastructure (HII)
- Uses UEFI specifications HII APIs to show/set UEFI settings
- Decoupled from shells

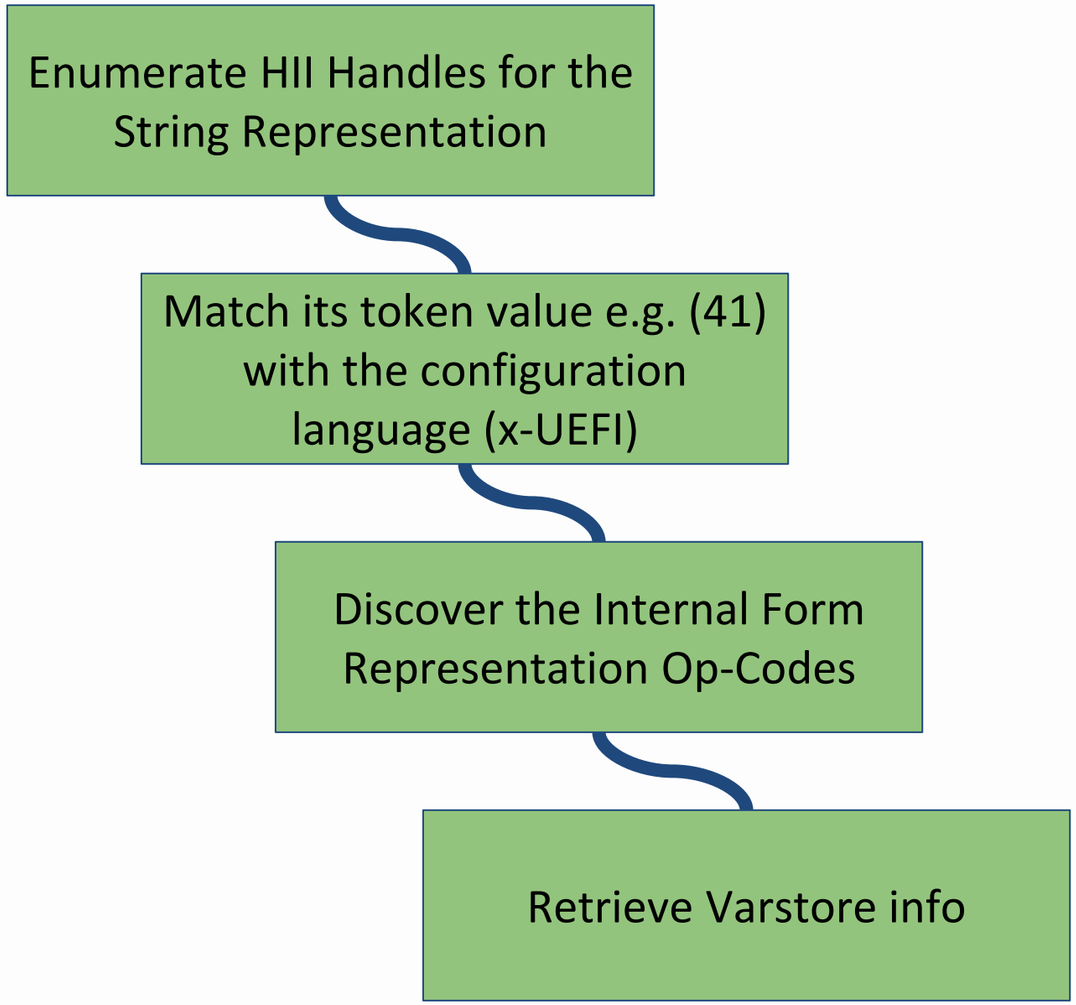


Figure 1: Mechanics of the vendor neutral tool. Adapted from https://uefi.org/namespace_instructions

Workflow

1. On power-on, UEFI firmware starts Boot Manager.

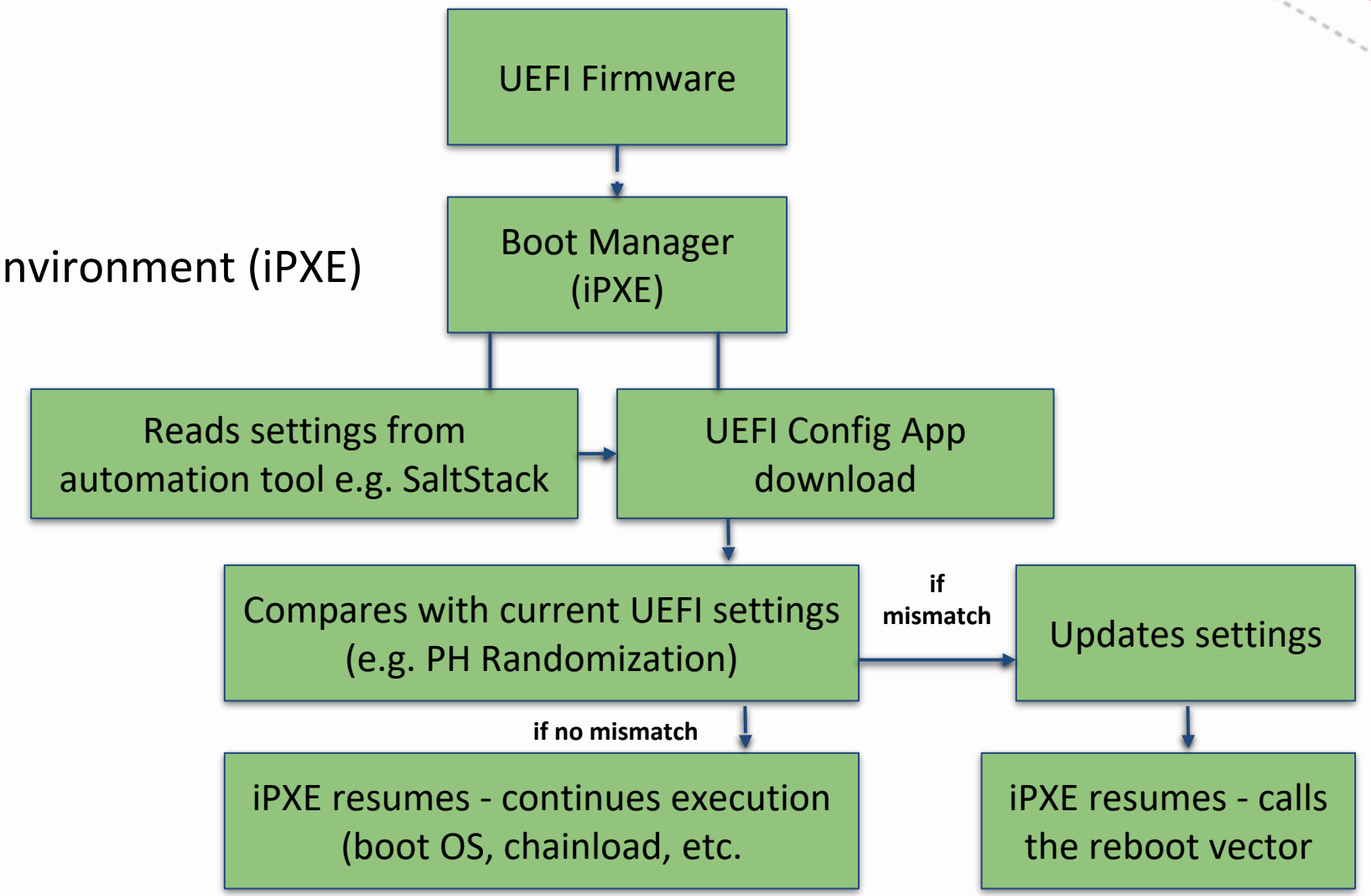
2. Boot Manager runs Preboot Execution Environment (iPXE)

3. Preboot Execution Environment downloads the UEFI Config App and reads the expected config from automation

4. Config app gets executed with the config data

5. UEFI Config app compares/updates settings, exits.

6. Preboot Execution Environment (iPXE) resumes and proceeds with boot logic.



Config Made Easy

/* setting the UEFI config in infrastructure management tool for a generation of server from a specific OEM */

```
<PRODUCT NAME OF SERVER>:
  settings:
    USB Mass Storage Driver Support: Disabled
    COM0-> Terminal Type: ANSI
    Media detect count: 2
    PH Randomization: Enabled
```

Config Made Easy

/* sample of templated generic loop version to **show** settings in infrastructure management tool */

```
:show_settings
{% for key, value in settings.items() %}
  HIIConfig show {{ key }}
  iseq ${HIIBufferVar} {{ value }} || goto set_settings
{% endfor %}
```

Config Made Easy

/* snippet of execution from show_settings */

```
Getting configuration “COM0->Terminal Type”
Type: EFI_IFR_ONE_OF_OP
OneofOption Option: "VT100" Value: 0
OneofOption Option: "VT100Plus" Value: 1
OneofOption Option: "VT-UTF8" Value: 2
OneofOption Option: "ANSI" Value: 3
…
Current config value is: “ANSI”
```

Config Made Easy

/* sample of templated generic loop version to **“set”** settings in infrastructure management tool
*/

```
:set_settings
{% for key, value in settings.items() %}
  HIIConfig set {{ key }} {{ value }} || goto exit_gracefully
{% endfor %}
```

Config Made Easy

/* snippet of execution from set_settings */

```
Setting Config: Media detect count=2
OpCode is 0x07
Type: EFI_IFR_NUMERIC_OP
MinValue      - 1
MaxValue      - 50
Step          - 1
...
GUID=165d40d1fc7a9546bb1241459d3695a2&...OFFSET=
0005&WIDTH=0001&VALUE=2
Success! Configuration change accepted
```

Integration With Baseboard Management Controller

- Exports configuration to Baseboard Management Controller via Intelligent Platform Management Interface/Redfish
- Can support Baseboard Management Controller in firmware update operations
- Redfish endpoints available for out-of-band BIOS configuration view
- Accessible variable lists, attributes, defaults
- BIOS configuration changes remain in PreBoot Execution Environment/UEFI shells

Key Takeaways

- Vendor-neutral UEFI configuration tool
- Handles complex configurations e.g. take in settings as lists
- Integrates with existing infrastructure configuration management tools e.g. Salt
- Intuitive to use
- Reduces human errors
- Eliminates vendor delays
- Ease of maintenance

<DEMO>

Questions?

References

- UEFI Namespace instructions: https://uefi.org/namespace_instructions
- The role of Redfish in UEFI forum firmware specifications https://uefi.org/sites/default/files/resources/UEFI%20Forum%20Redfish%20Webinar_website%20Final.pdf