tianocore

UEFI & EDK II TRAINING

UEFI Human Interface Infrastructure (HII)

tianocore.org

tianocore

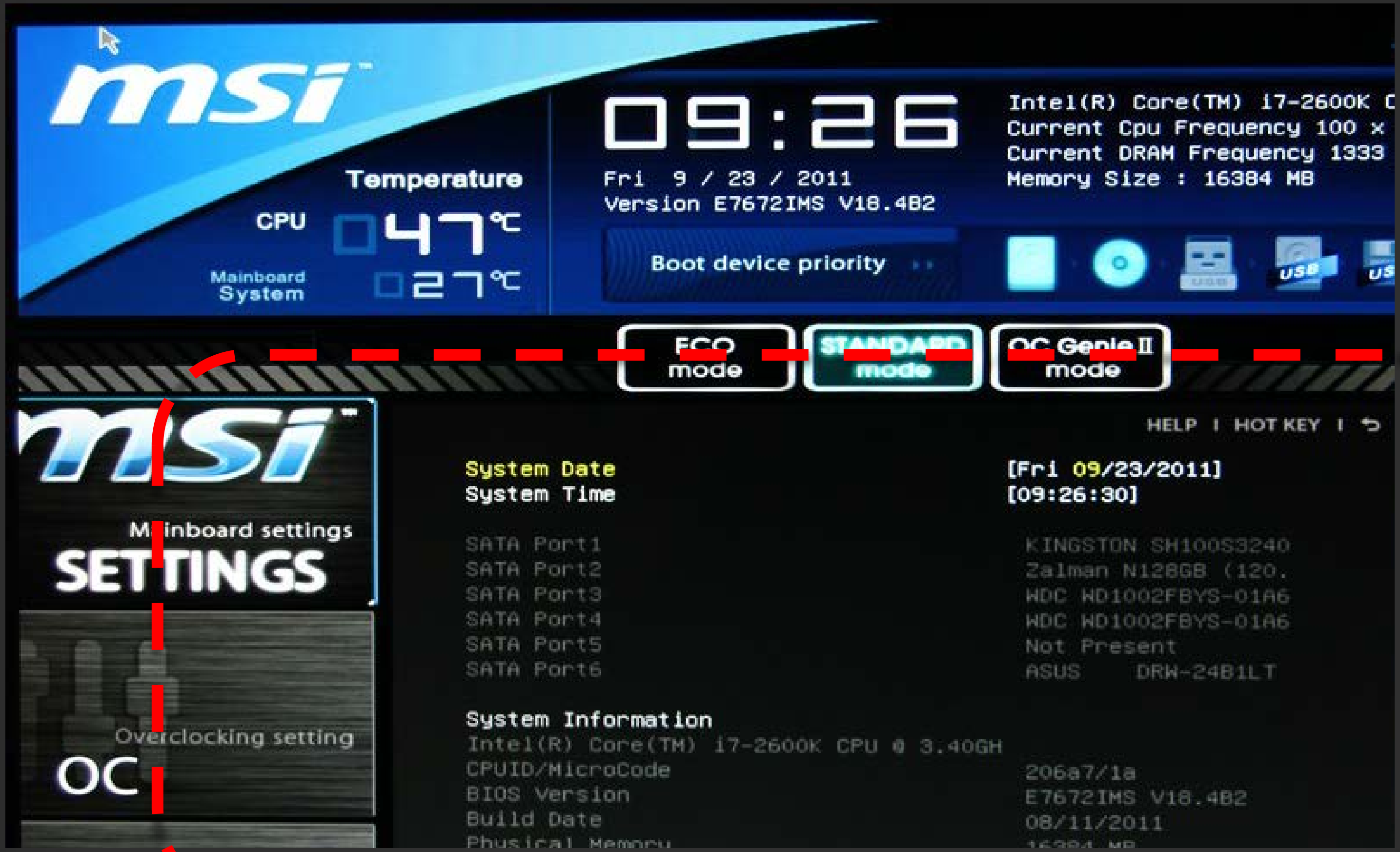
Lesson Objective

- What is the Infrastructure for HII
- How Does HII Work
- Lab for HII

USER INTERFACE HII OVERVIEW

tianocore

Why ?



Temperature
CPU 47℃
Mainboard System 27℃

```
09:26
Fri 9 / 23 / 2011
Version E7672IMS V18.4B2
```

```
Intel(R) Core(TM) i7-2600K C
Current Cpu Frequency 100 x
Current DRAM Frequency 1333
Memory Size : 16384 MB
```

Boot device priority

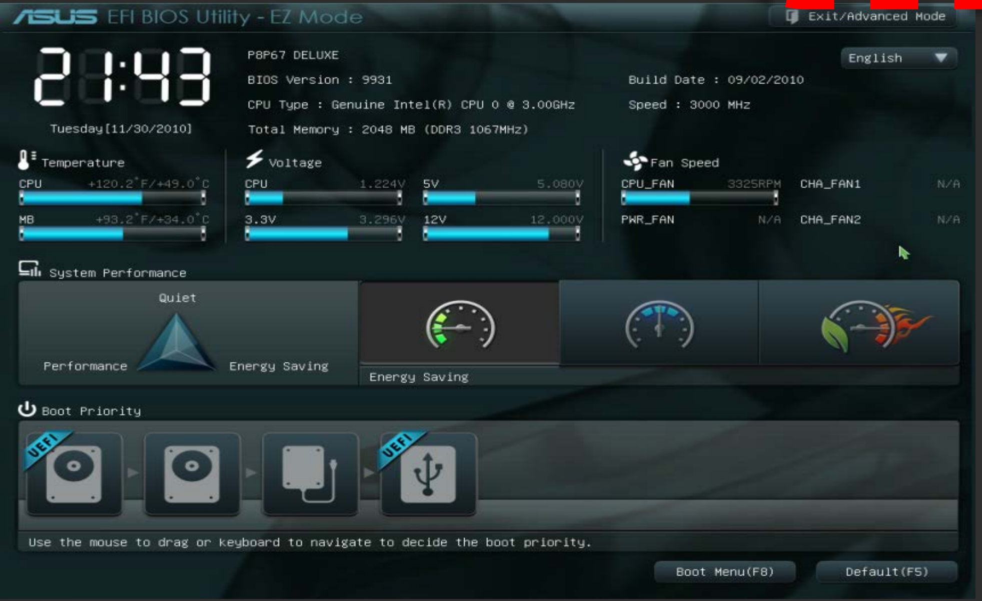
ECO mode | STANDARD mode | OC Genie II mode

HELP | HOT KEY

Mainboard settings
SETTINGS

Overclocking setting
OC

```
System Date            [Fri 09/23/2011]
System Time            [09:26:30]

SATA Port1             KINGSTON SH100S3240
SATA Port2             Zalman N128GB (120.
SATA Port3             WDC WD1002FBYS-01A6
SATA Port4             WDC WD1002FBYS-01A6
SATA Port5             Not Present
SATA Port6             ASUS    DRW-24B1LT

System Information
Intel(R) Core(TM) i7-2600K CPU @ 3.40GH
CPUID/MicroCode        206a7/1a
BIOS Version           E7672IMS V18.4B2
Build Date             08/11/2011
Physical Memory        16384 MB
```



EFI BIOS Utility - EZ Mode

Exit/Advanced Mode

```
21:43                   P8P67 DELUXE                                           English
                        BIOS Version : 9931                        Build Date : 09/02/2010
                        CPU Type : Genuine Intel(R) CPU 0 @ 3.00GHz  Speed : 3000 MHz
Tuesday[11/30/2010]     Total Memory : 2048 MB (DDR3 1067MHz)

Temperature                 Voltage                               Fan Speed
CPU  +120.2°F/+49.0°C       CPU   1.224V    5V    5.080V          CPU_FAN  3325RPM   CHA_FAN1  N/A
MB   +93.2°F/+34.0°C        3.3V  3.296V    12V   12.000V         PWR_FAN  N/A       CHA_FAN2  N/A
```

System Performance

Quiet
Performance
Energy Saving

Energy Saving

Boot Priority

Use the mouse to drag or keyboard to navigate to decide the boot priority.

Boot Menu(F8) | Default(F5)

Unified Look and Feel at Platform level

Single Interface

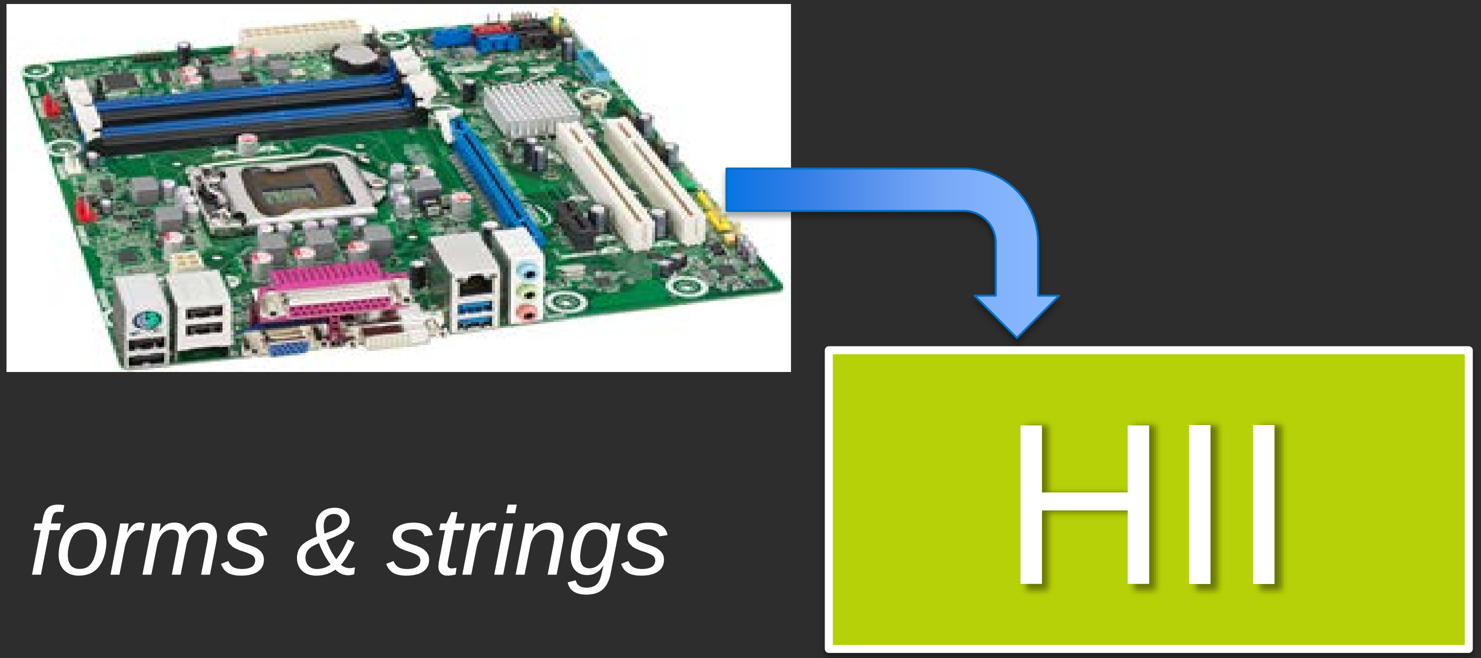
Localization

• • •

HII: Key Concepts

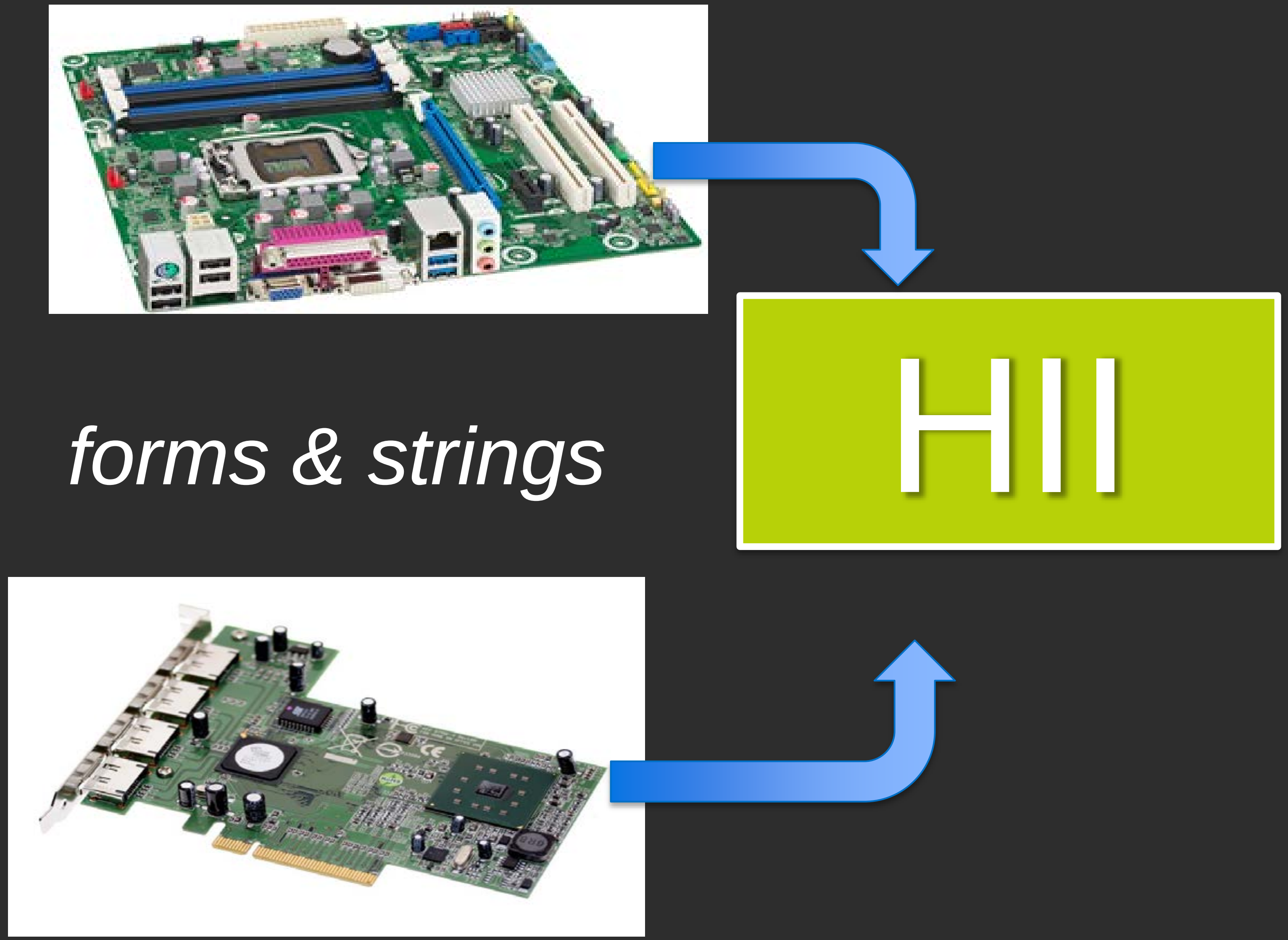
forms & strings

HII: Key Concepts

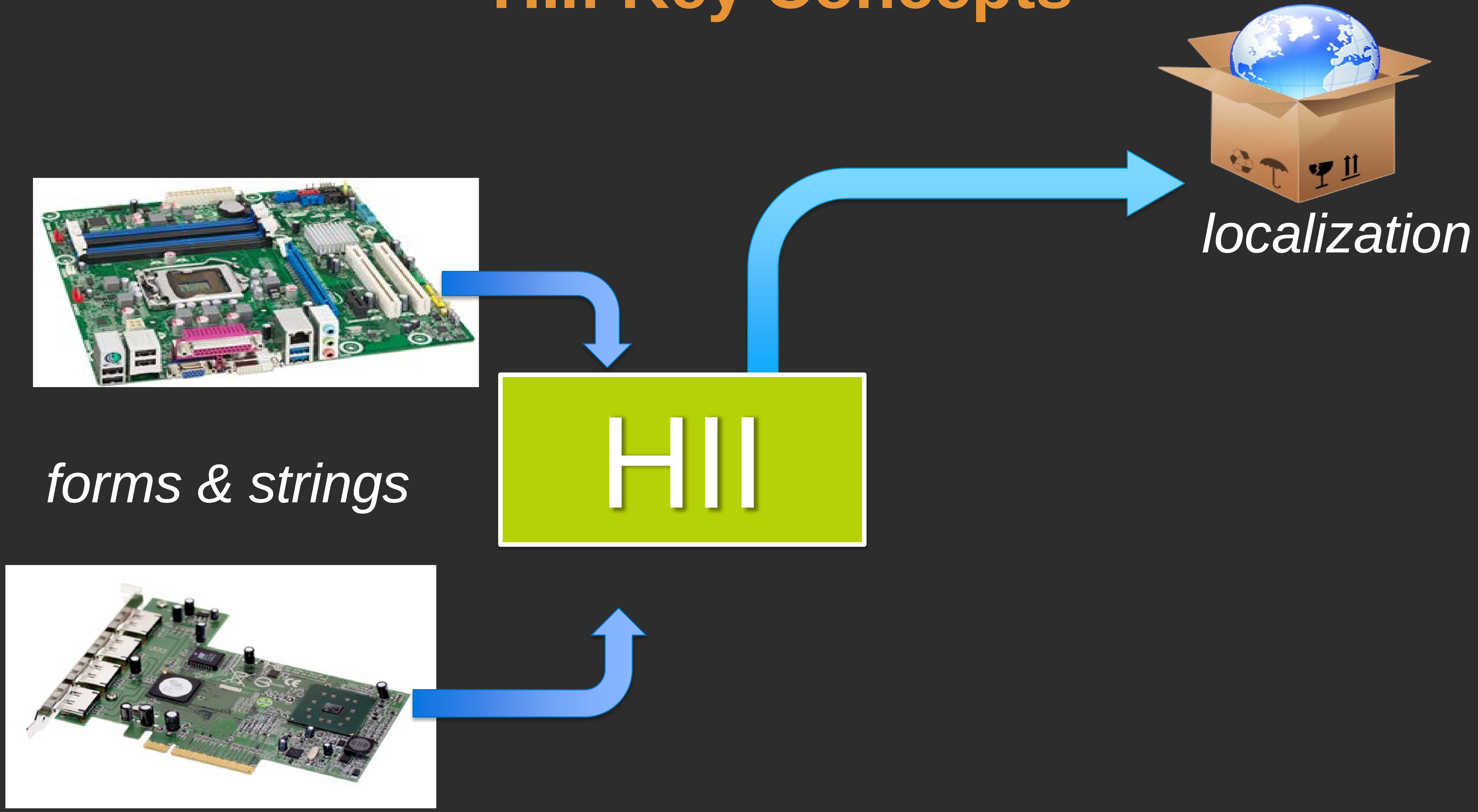


forms & strings

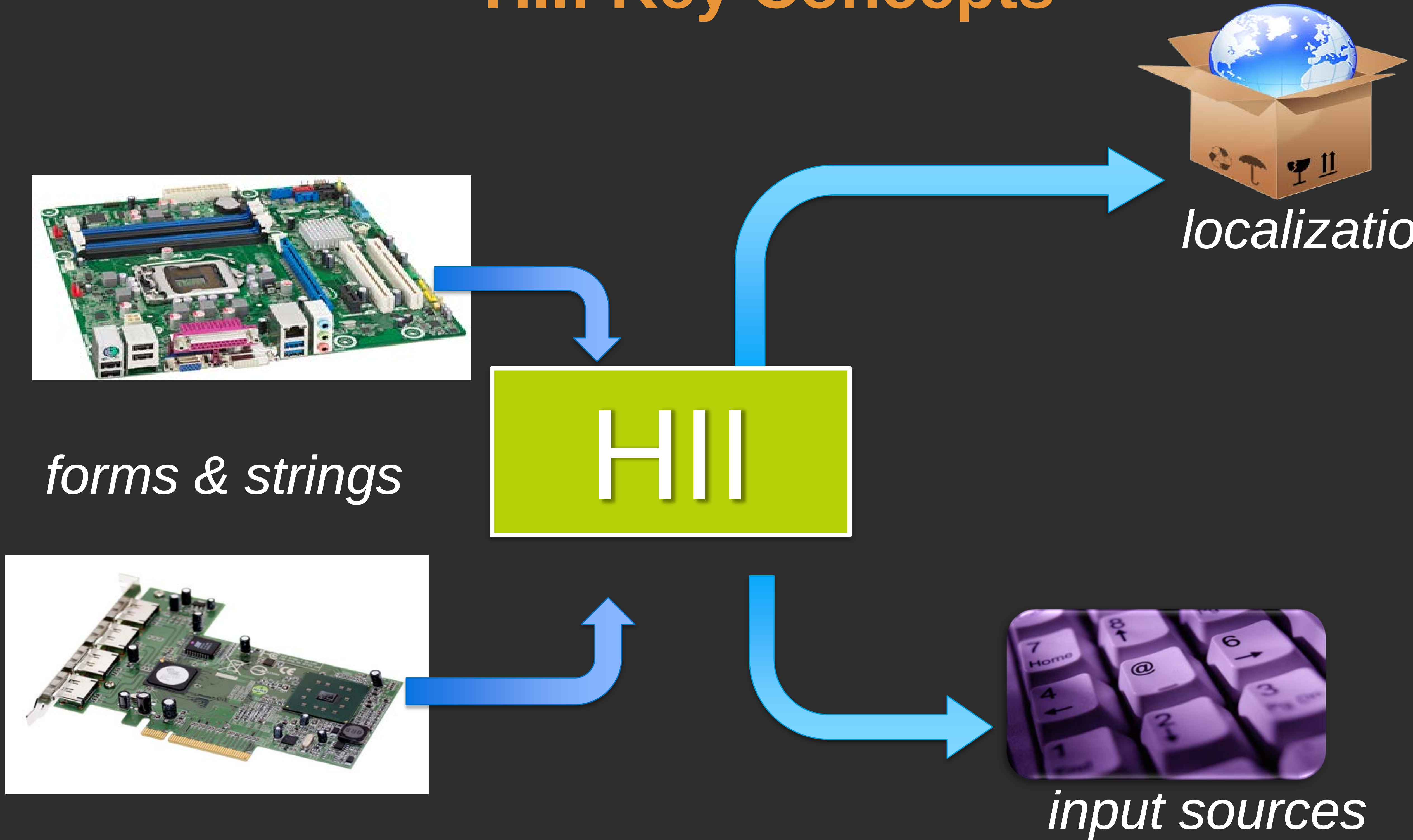
HII: Key Concepts



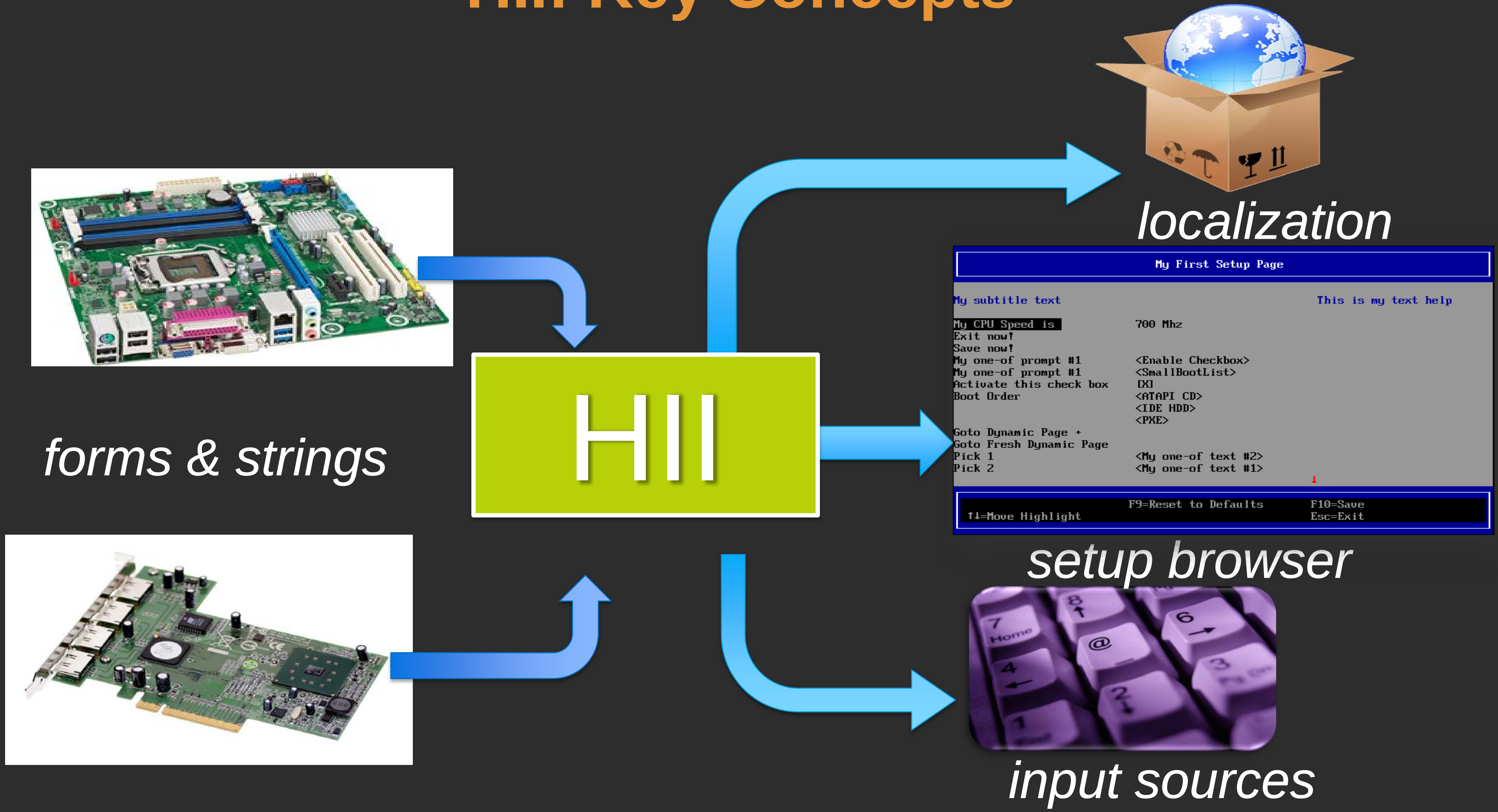
HII: Key Concepts

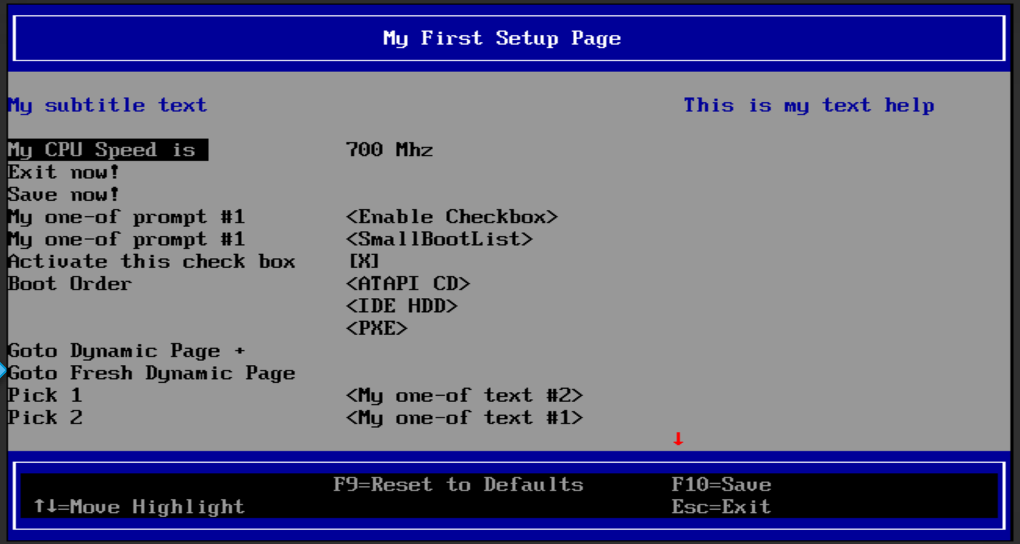


HII: Key Concepts



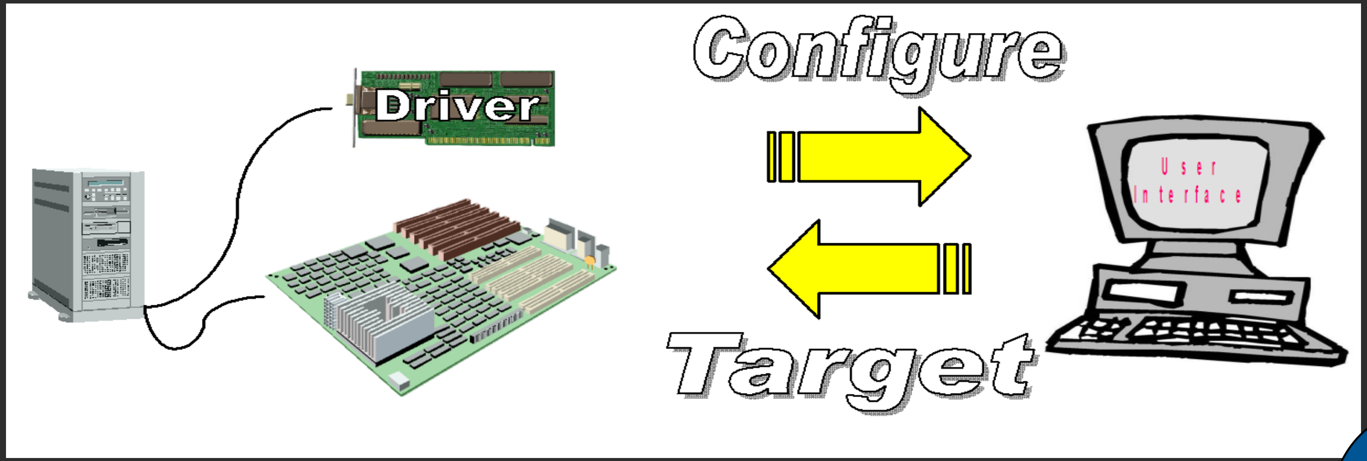
HII: Key Concepts

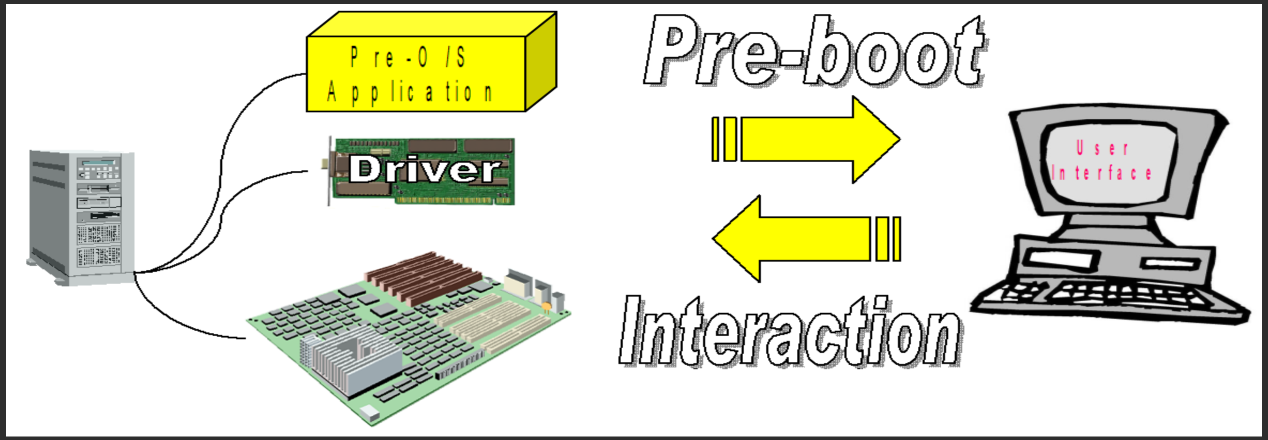


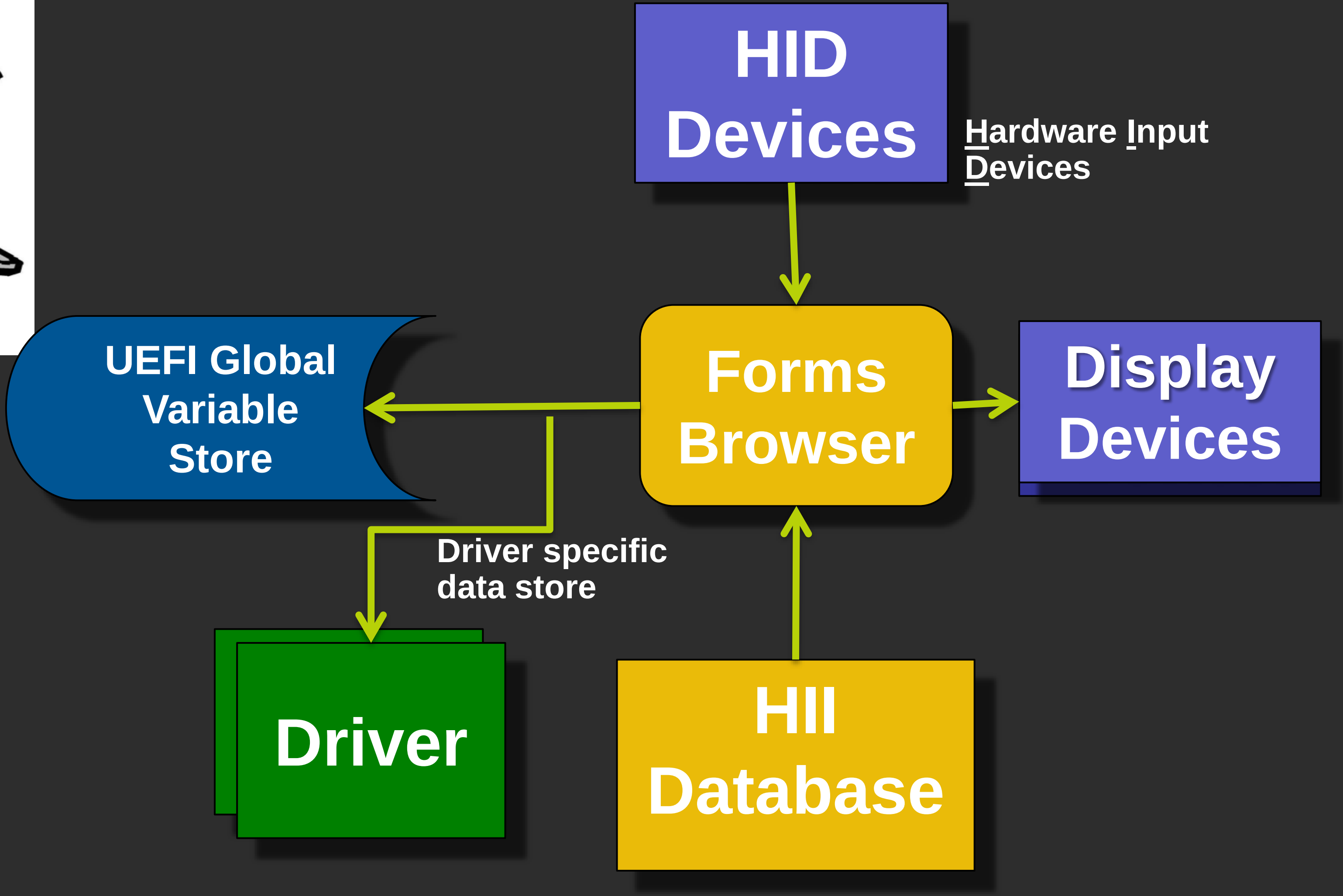


```
                         My First Setup Page

My subtitle text                                         This is my text help

My CPU Speed is              700 Mhz
Exit now!
Save now!
My one-of prompt #1          <Enable Checkbox>
My one-of prompt #1          <SmallBootList>
Activate this check box      [X]
Boot Order                   <ATAPI CD>
                             <IDE HDD>
                             <PXE>
Goto Dynamic Page +
Goto Fresh Dynamic Page
Pick 1                       <My one-of text #2>
Pick 2                       <My one-of text #1>

                        F9=Reset to Defaults          F10=Save
  ↑↓=Move Highlight                                   Esc=Exit
```

Design Discussions







See § 29.2 of the UEFI 2.x Spec.

HII COMPONENTS

tianocore

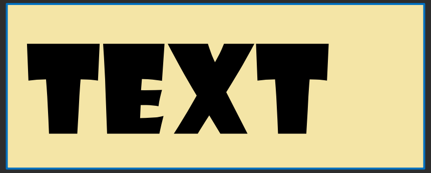
Human Interface Components

Strings

TEXT

Human Interface Components

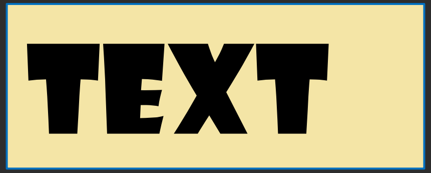
Strings



Fonts

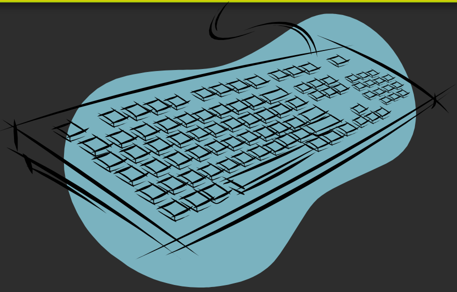
Human Interface Components

Strings



Fonts

Keyboard

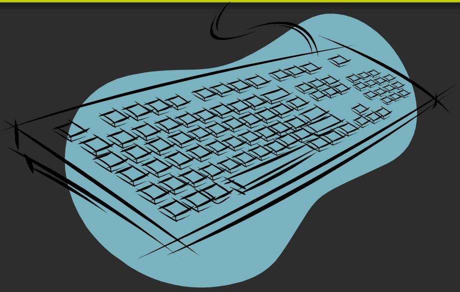


Human Interface Components

Strings

Fonts

Keyboard



Forms

Human Interface Components

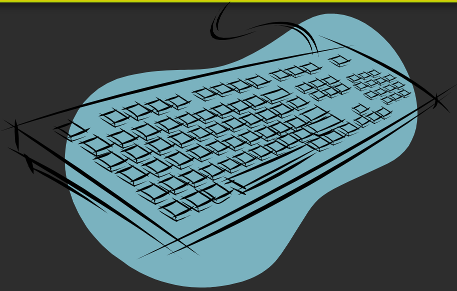
Strings

TEXT

Fonts

AB前

Keyboard



Forms

Packages

Strings

Strings stored in Unicode

- Real string encodings required for e.g. VT100
- Already the text standard in UEFI today

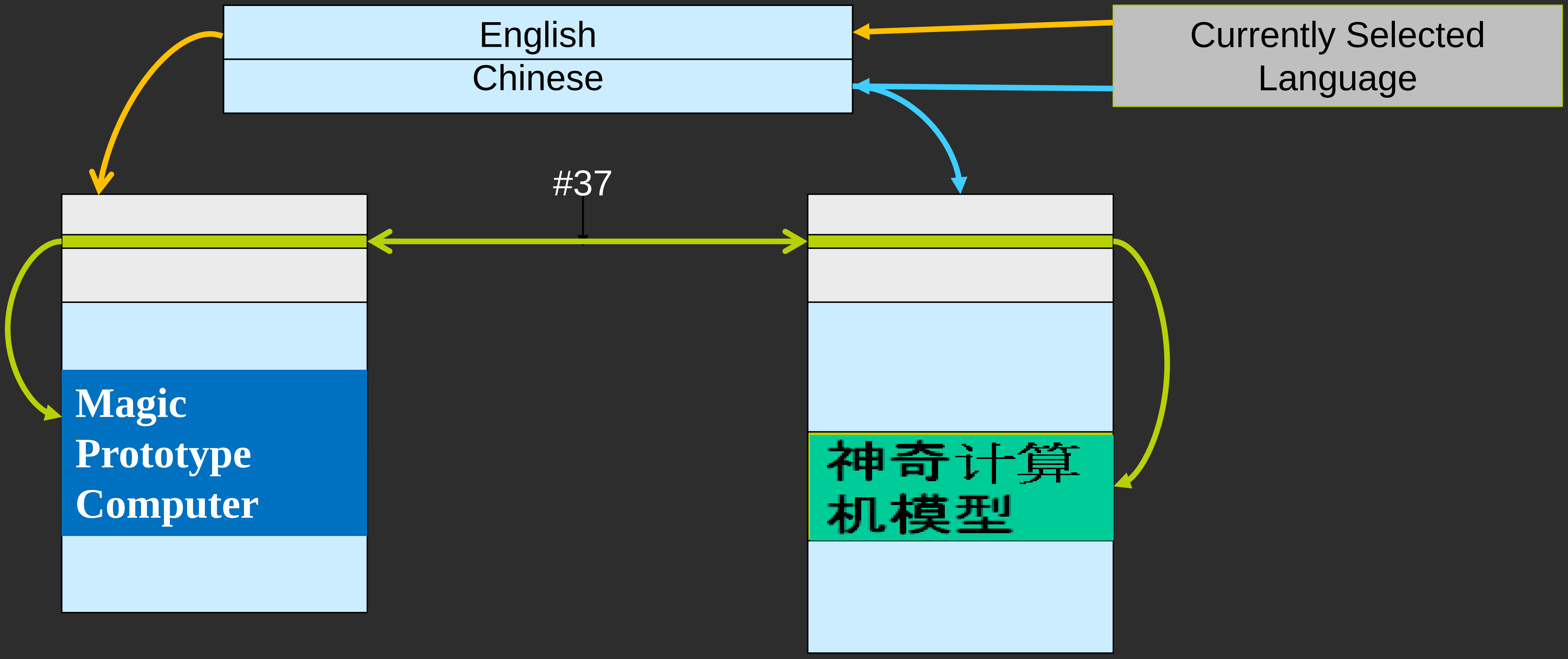
Localization happens at the string level

- Caller externs and passes in language independent string token
- String support determines actual string from token and selected language
- Usage Model:
  - A string library supporting translations
    - Reduces translation costs and delays
  - Tools to extract strings depending on use by driver
  - Analysis of strings used to extract fonts
  - RFC 4646 Language codes (2-2)

tianocore

Token to String Mapping

Request: Print string with token 37
Currently selected language is as in UEFI 2.X. This is used to select between language data structures. (The structures indicate which language(s) they support).
The top part of the structure maps from token to string. The bottom part of the structure is the strings



tianocore

String Example (.uni file)

Source code

```
#langdef      en-US    "English"
#langdef      fr-FR    "Francais“
#langdef      sv-SE    “Svenska”

#string STR_FORM_SET_TITLE
#language en-US "Browser Testcase Engine"
#language fr-FR "Navigateur Testcase Moteur”
#language sv-SE "Webbläsare Testcase Motor”

#string STR_FORM_SET_TITLE_HELP
#language en-US "This is a sample UEFI driver which is
used to test the browser op-code operations. "
#language fr-FR “Il s'agit d'une UEFI Driver
échantillon qui est utilisé pour tester les navigateurs
op-code opérations.“
#language sv-SE ”Detta är ett exempel på UEFI-drivrutin
som används för att testa webbläsaren op-kod
operationer”

#string STR_FORM1_TITLE
#language en-US "My First Setup Page"
#language fr-FR "Mi Primero Arreglo Página“
#language sv-SE "Min första inställningssidan”
```

RFC 4646 Language codes

Fonts

One Standard Font for UEFI

- One font database accumulated during boot

Each Component Provides Its Fonts

- System provides ASCII and ISO Latin-1
- Fonts only required for characters in strings that may appear
  - If the firmware will never print “tractor” in Kanji, discard the bit image
- Result is a sparse array of characters indexed by the Unicode ‘weight’

Wide and Narrow glyphs supported

A

A

B

前

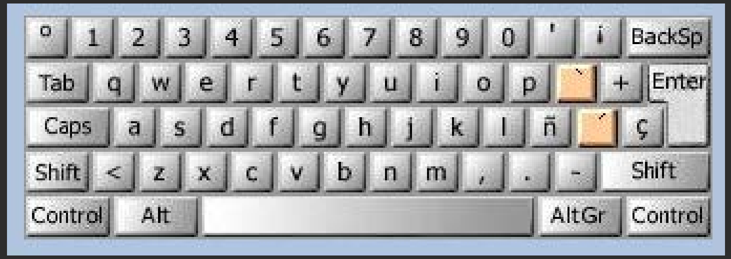
Keyboards

Support varying keyboards

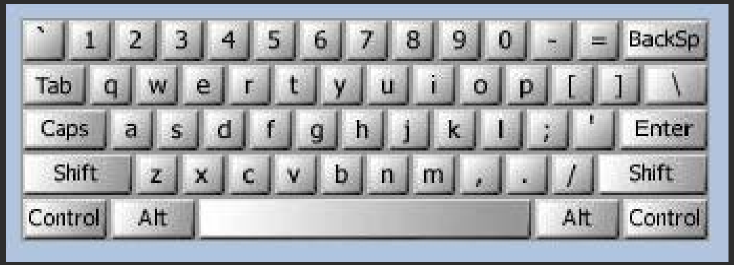
- UK and US keyboard layout are not the same. Certainly, that is the case for US and Arabic, etc.
- Adding support of other modifiers (e.g. Alt-GR, Dead-keys, etc)

Keyboard Layout

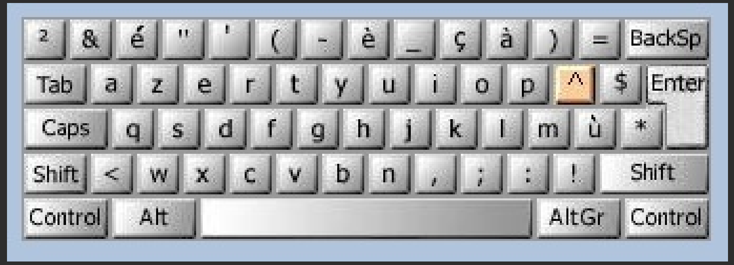
- Allow for a standardized mechanism to describe a keyboard layout and add to system database.
- Allow for switching keyboard layouts.



Spanish

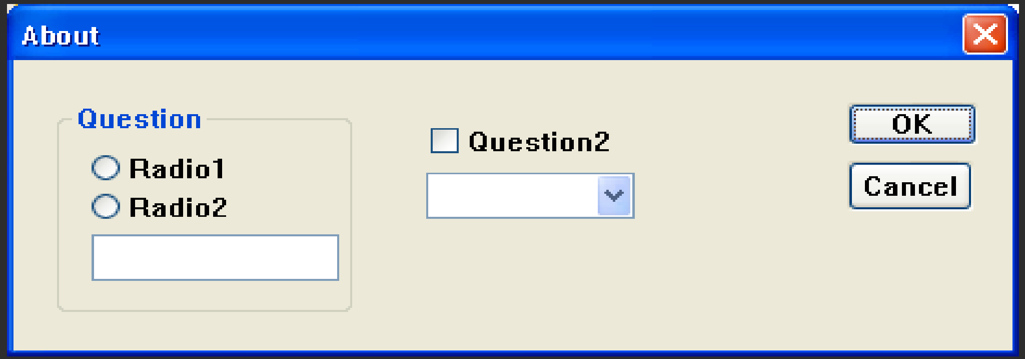


English



French

Forms



- The forms are stored in the HII database, along with the strings, fonts & images
- Other applications may use the information within the forms to validate configuration setting values
- The Forms Browser provides a forms-based user interface which understands
  - how to read the contents of the forms
  - interact with the user
  - save the resulting values
- The Forms Browser uses forms data installed by an application or driver during initialization in the HII database.

tianocore

Visual Forms Representation (VFR)

- Language used to describe what a page layout would be in a browser as well as the op-codes and string tokens to display
- Op-codes are defined for the following functions examples
  - **`formSet`** and **`form`** definitions
  - One of type questions with corresponding options (combo) fields

```
checkbox
numeric
oneof
String
```

  - Boolean expressions in support of errors, suppress, and gray outs

```
"disableif"
“suppressif“
“grayoutif"
```

Form Example (.vfr file)

```
formset
  guid      = FORMSET_GUID,
  title     = STRING_TOKEN(STR_FORM_SET_TITLE),
  help      = STRING_TOKEN(STR_FORM_SET_TITLE_HELP),
  classguid = EFI_HII_PLATFORM_SETUP_FORMSET_GUID,

 varstore   DRIVER_SAMPLE_CONFIGURATION,
      name = MyIfrNVData,
      guid = FORMSET_GUID;

 form formid = 1,
       title  = STRING_TOKEN(STR_FORM1_TITLE);

 oneof varid  = MyIfrNVData.MyVariableForOneofPrompt,
  prompt      = STRING_TOKEN(STR_ONE_OF_PROMPT),
   help       = STRING_TOKEN(STR_ONE_OF_HELP),
  option text = STRING_TOKEN(STR_ONE_OF_TEXT1), value = 0x0, flags = 0;
  option text = STRING_TOKEN(STR_ONE_OF_TEXT2), value = 0x1, flags = 0;
  option text = STRING_TOKEN(STR_ONE_OF_TEXT3), value = 0x2, flags = DEFAULT;
 endoneof;
                                    • • •

 endform;
endformset;
```

Source code

tianocore

Internal Forms Representation (IFR)

- IFR Code created by VFR to IFR compiler tool
- Byte encoded operations (much smaller)
- String references abstracted as tokens
- Improved validation, visibility primitives
- At better level of presentation control for firmware
  - Tension between configuration driver and presentation driver over control of presentation format

Easy to

- Interpret for small Setup engine in desktop firmware
- Translate into XHTML or JavaScript or …

tianocore

Minimum Files for HII Driver

.c

.h

tianocore

Minimum Files for HII Driver

.c

.h

.uni

.vfr

tianocore

Minimum Files for HII Driver

.c

.h

.uni

.vfr

.inf

tianocore

Minimum Files for HII Driver

.c

.h

.uni

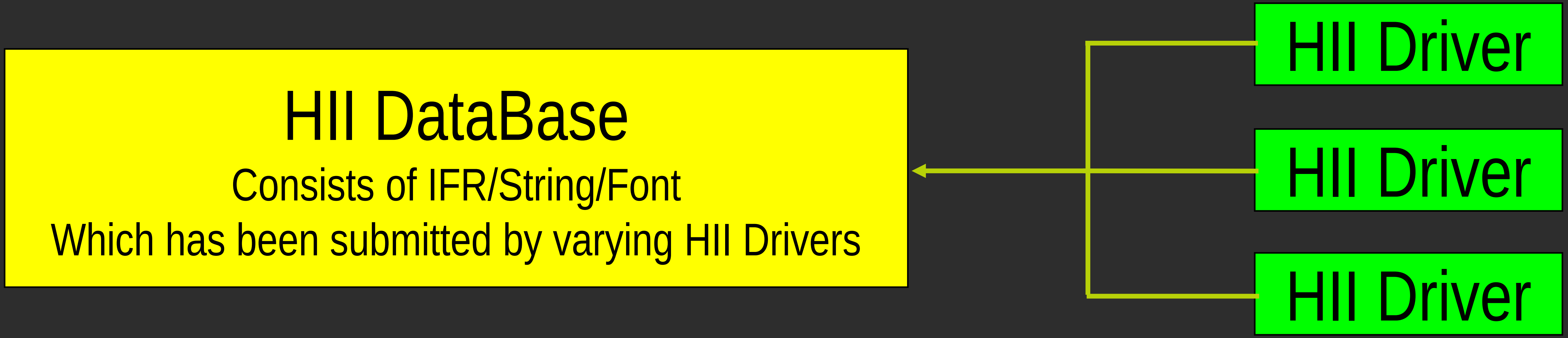
Strings

.vfr

Forms

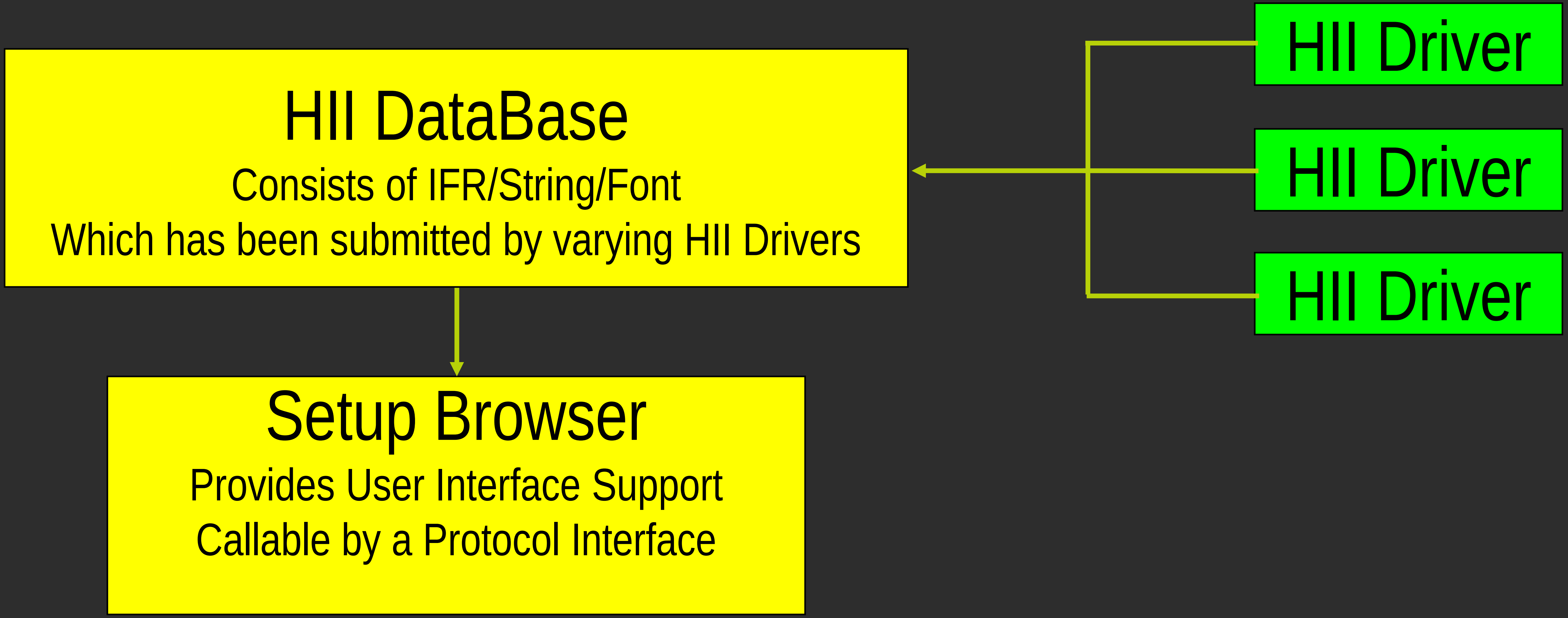
.inf

EDK II HII


HII DataBase
Consists of IFR/String/Font
Which has been submitted by varying HII Drivers
HII Driver
HII Driver
HII Driver

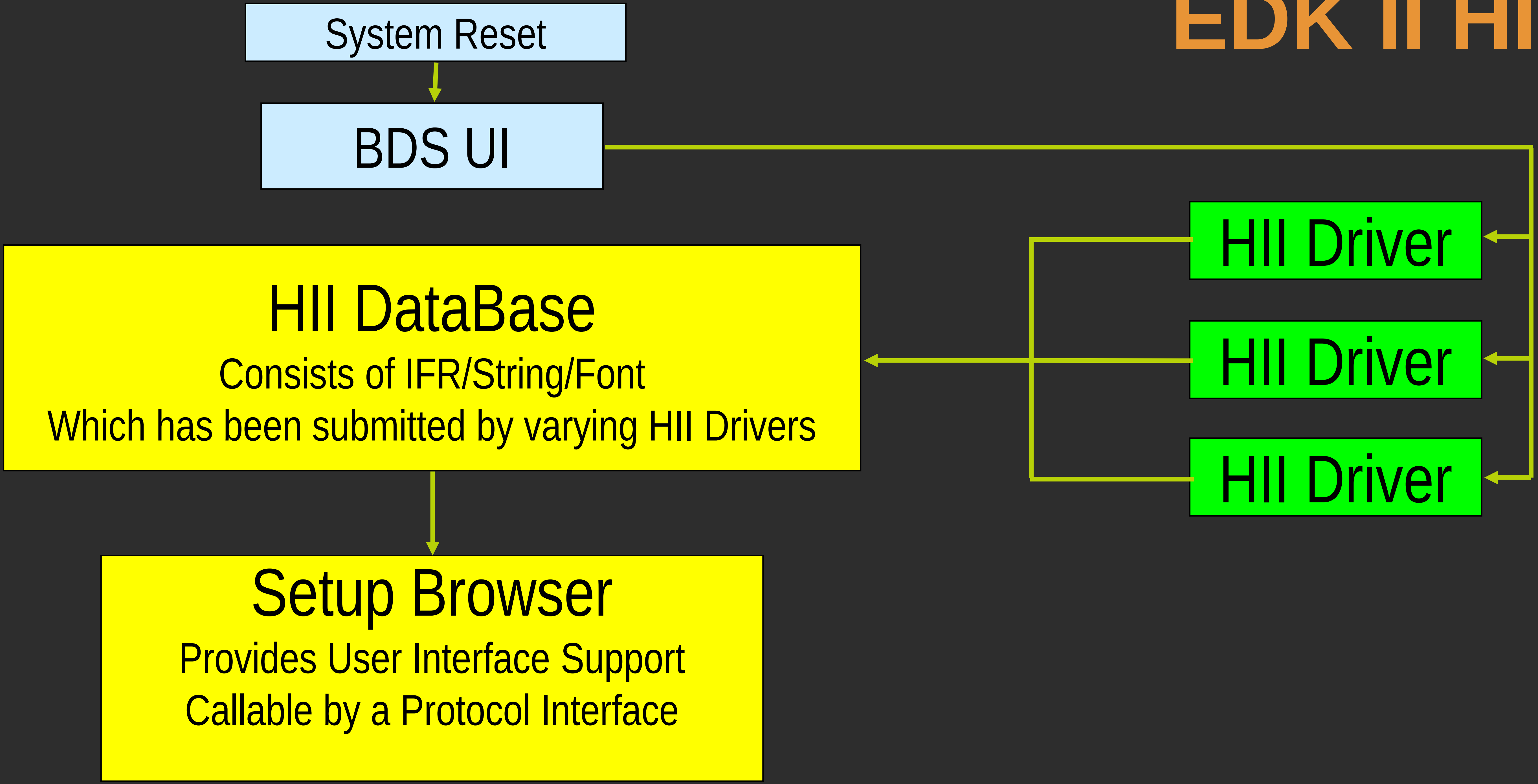
tianocore

EDK II HII

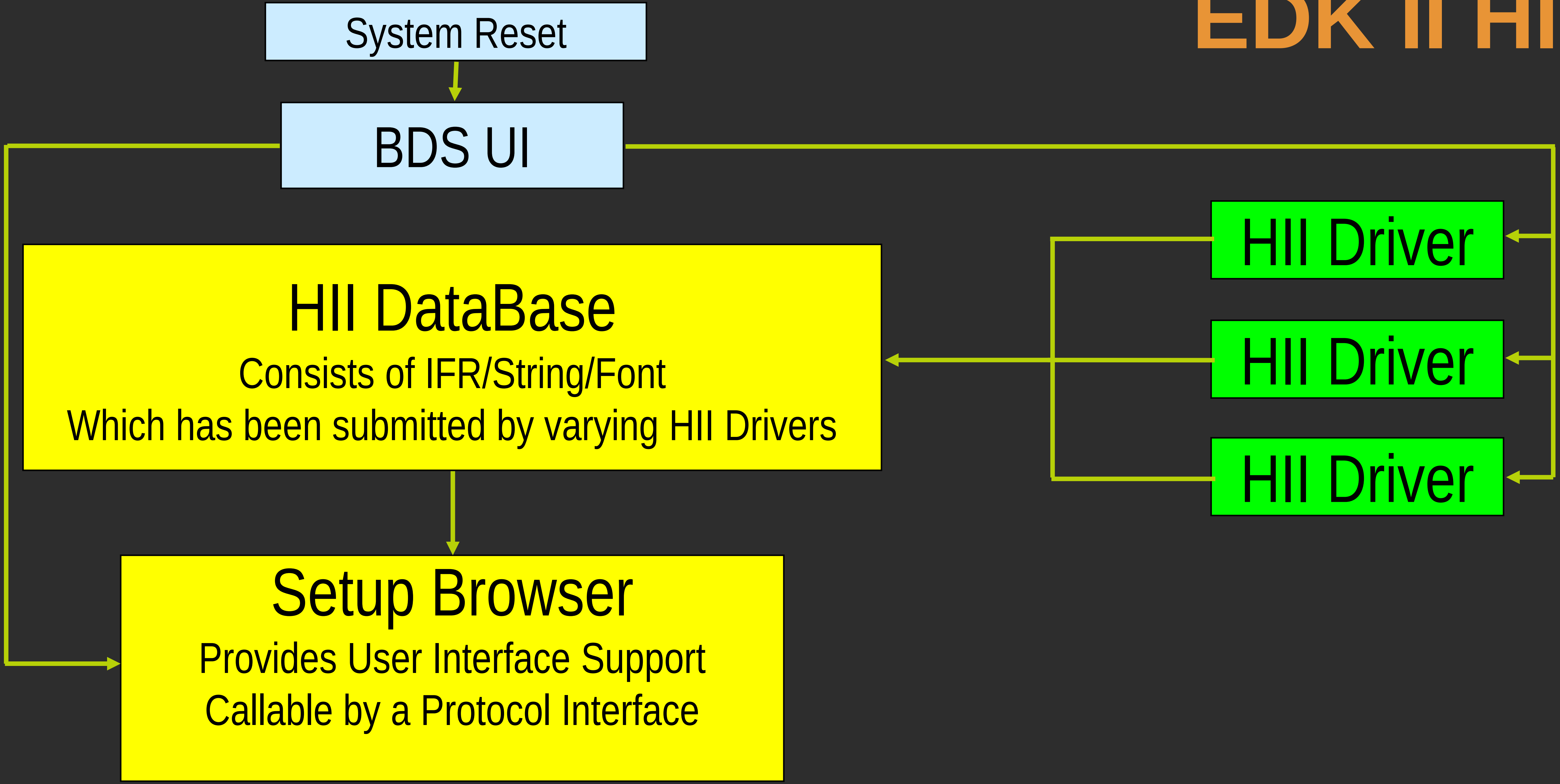

HII DataBase
Consists of IFR/String/Font
Which has been submitted by varying HII Drivers
Setup Browser
Provides User Interface Support
Callable by a Protocol Interface
HII Driver
HII Driver
HII Driver

tianocore

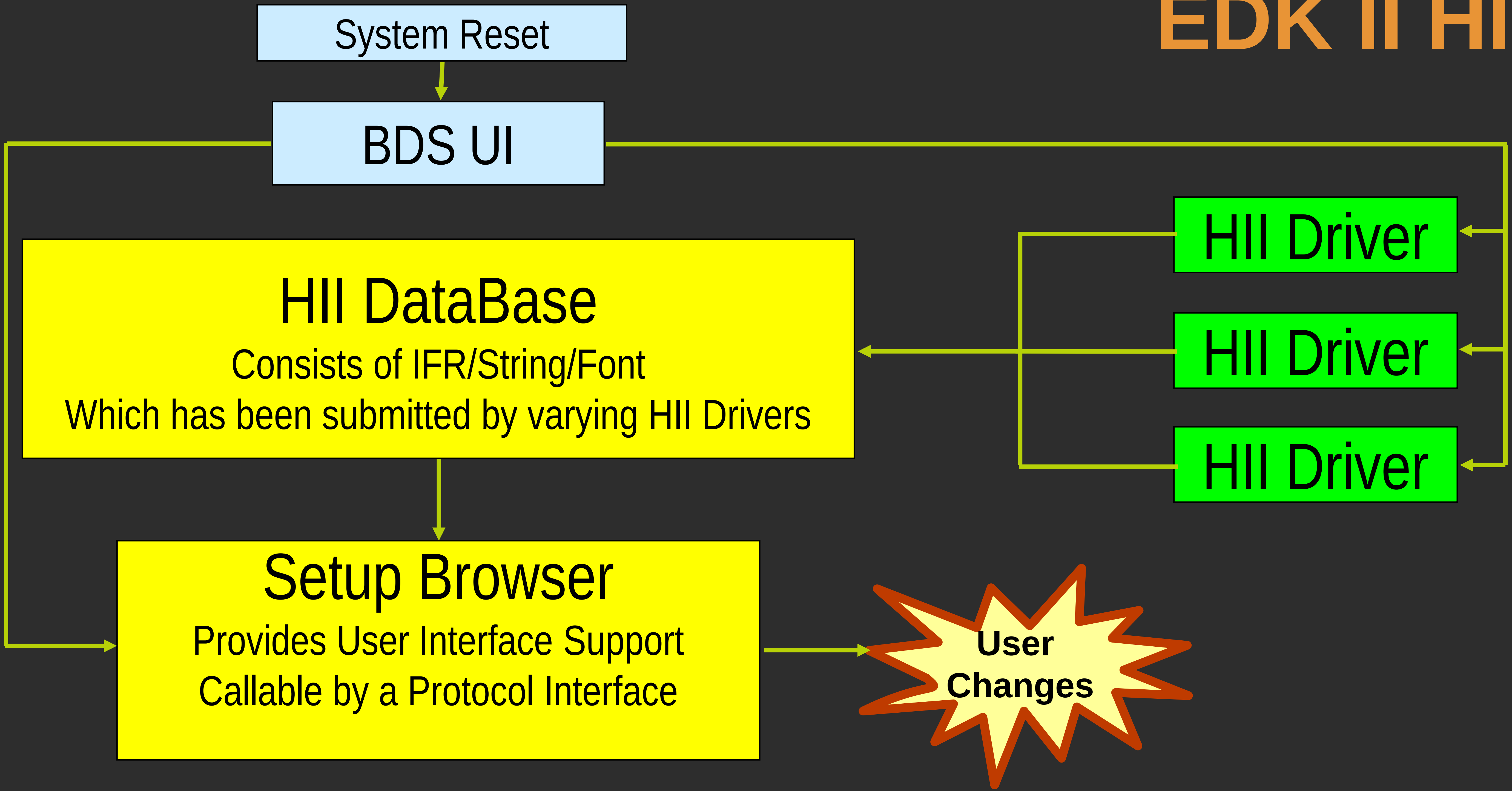
EDK II HII


System Reset
BDS UI
HII DataBase
Consists of IFR/String/Font
Which has been submitted by varying HII Drivers
HII Driver
HII Driver
HII Driver
Setup Browser
Provides User Interface Support
Callable by a Protocol Interface

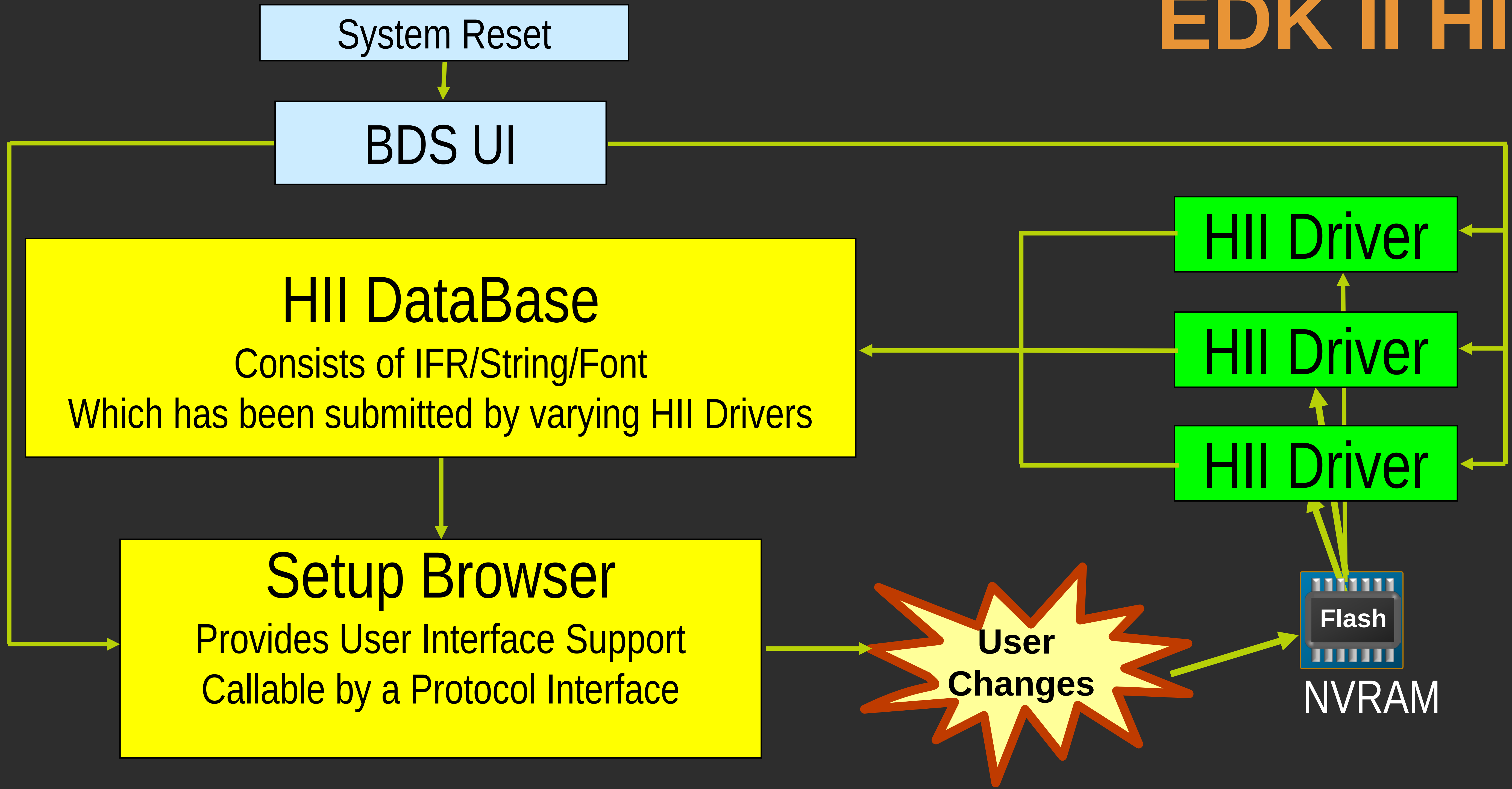
EDK II HII


System Reset
BDS UI
HII DataBase
Consists of IFR/String/Font
Which has been submitted by varying HII Drivers
HII Driver
HII Driver
HII Driver
Setup Browser
Provides User Interface Support
Callable by a Protocol Interface

EDK II HII


System Reset
BDS UI
HII DataBase
Consists of IFR/String/Font
Which has been submitted by varying HII Drivers
HII Driver
HII Driver
HII Driver
Setup Browser
Provides User Interface Support
Callable by a Protocol Interface
User Changes

EDK II HII


System Reset
BDS UI
HII DataBase
Consists of IFR/String/Font
Which has been submitted by varying HII Drivers
Setup Browser
Provides User Interface Support
Callable by a Protocol Interface
HII Driver
HII Driver
HII Driver
User Changes
Flash
NVRAM

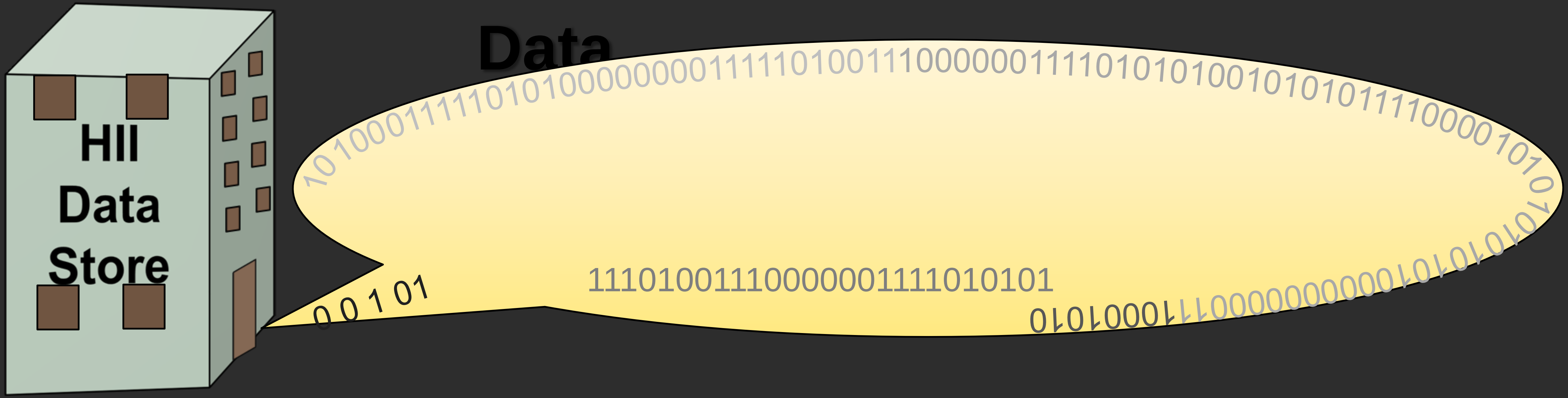
tianocore

HOW: UEFI HII PROTOCOLS

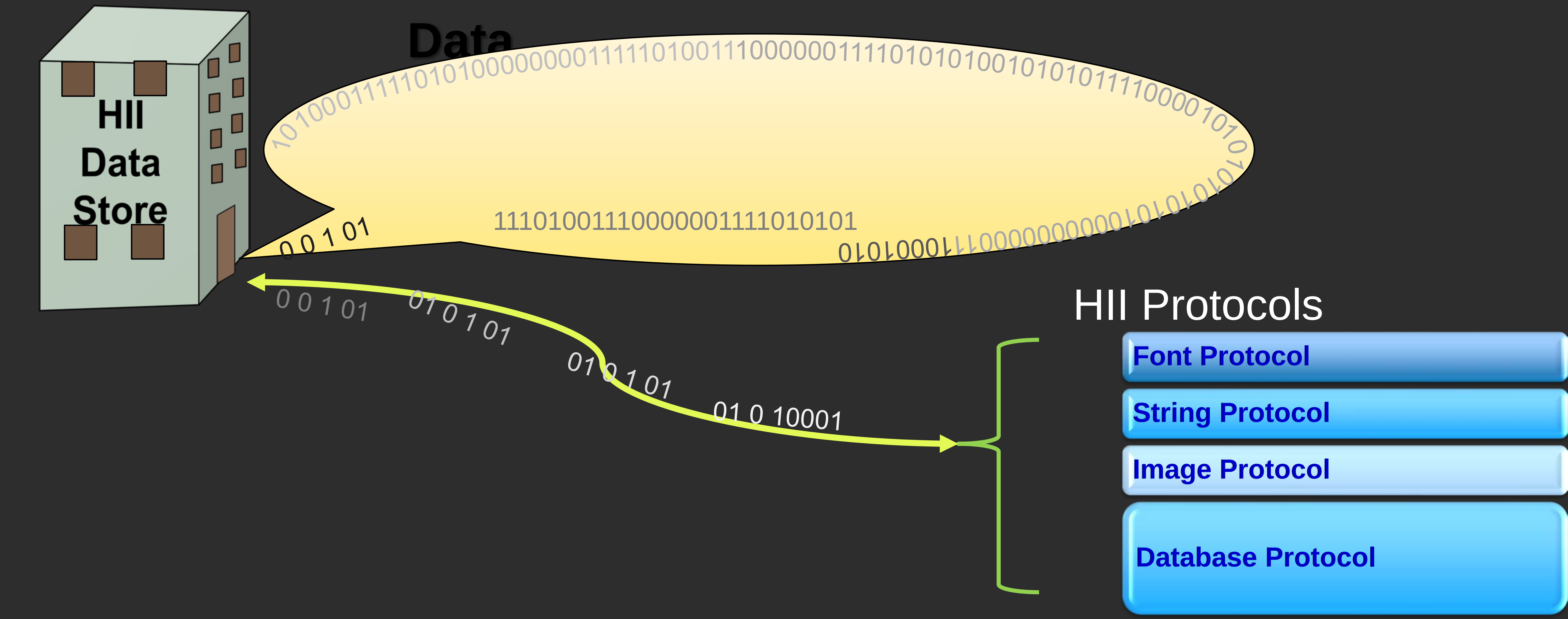
Sections 29-31 the UEFI 2.x Specification

HII Database Overview

HII Database Overview

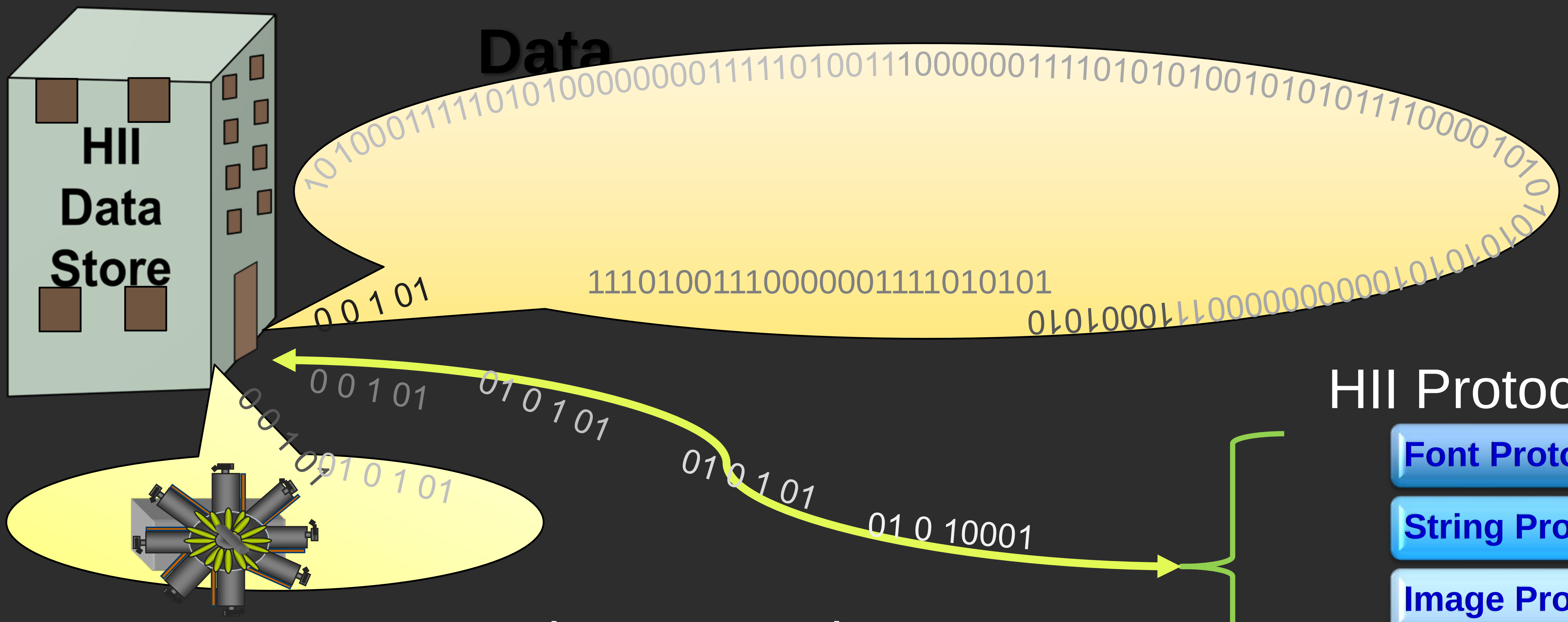


HII Database Overview



See § 30 of the UEFI 2.x Spec.

HII Database Overview



HII Protocols

- Font Protocol
- String Protocol
- Image Protocol
- Database Protocol

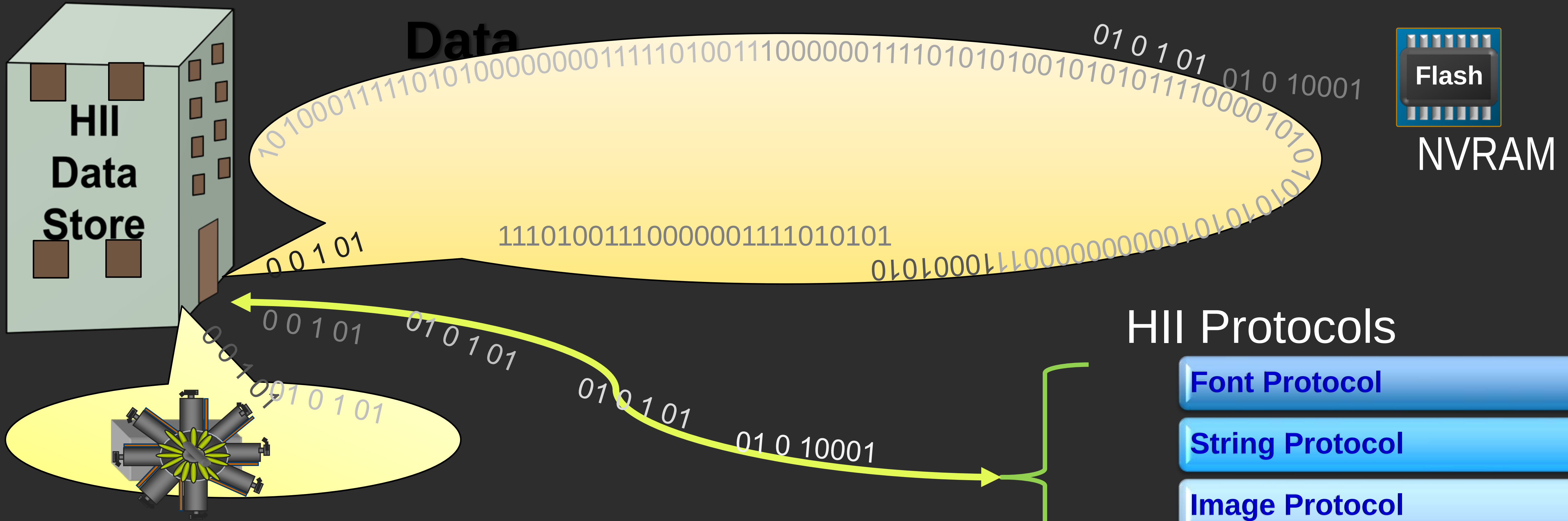
See § 30 of the UEFI 2.x Spec.

HII Browser Engine Protocols

- Configuration Routing Protocol
- Configuration Access Protocol
- Form Browser 2 Protocol

See § 31 of the UEFI 2.x Spec.

HII Database Overview



HII Protocols

- Font Protocol
- String Protocol
- Image Protocol
- Database Protocol

HII Browser Engine Protocols

- Configuration Routing Protocol
- Configuration Access Protocol
- Form Browser 2 Protocol

See § 31 of the UEFI 2.x Spec.

See § 30 of the UEFI 2.x Spec.

tianocore

UEFI HII Protocols

Font Protocol	• String to Image, Sting ID to Image, Get Glyph, Get Font Info
String Protocol	• New – Get – Set – String • Get Language & 2nd Language
Image Protocol	• New – Get – Set Image • Draw Image, Draw Image ID
Database Protocol	• New – Remove- Update – List – Export Lists – Get Handle Package • Register, Unregister Package Notify • Find- Get- Set Keyboard layout

See § 30 of the UEFI 2.x Spec.

UEFI Driver Initialization Process

HII Protocols

Config Routing Protocol
- ExtractConfig
- RouteConfig
- ExportConfig
- BlockToConfig
- ConfigToBlock

Form Browser 2 Protocol
- SendForm
- BrowserCallback

HII Database Protocols
- NewPackageList
- Remove
- Update
- . . .
- GetPackageListHandle

MyDriver

UEFI 2.x+ Driver
(e.g. Motherboard Driver, Addin card Op ROM)

Config Access Protocol

tianocore

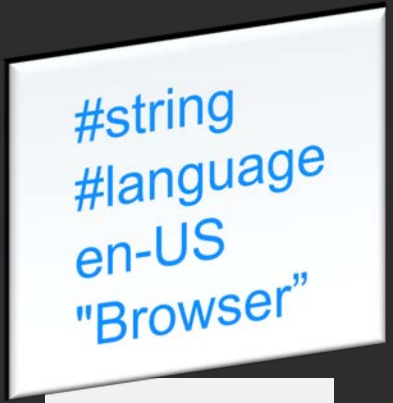
UEFI Driver Initialization Process

HII Protocols	
Config Routing Protocol	
	ExtractConfig
	RouteConfig
	ExportConfig
	BlockToConfig
	ConfigToBlock
Form Browser 2 Protocol	
	SendForm
	BrowserCallback
HII Database Protocols	
	NewPackageList
	Remove
	Update
	. . .
	GetPackageListHandle

MyDriver

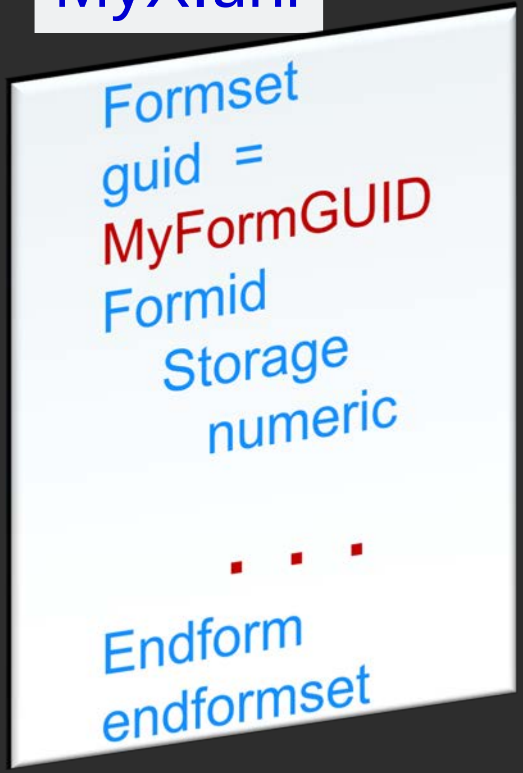
UEFI 2.x+ Driver
(e.g. Motherboard Driver, Addin card Op ROM)

Config Access Protocol



```
#string
#language
en-US
"Browser"
```

MyX.uni



```
Formset
guid  =
MyFormGUID
Formid
   Storage
     numeric

          . . .

Endform
endformset
```

MyVfr.vfr

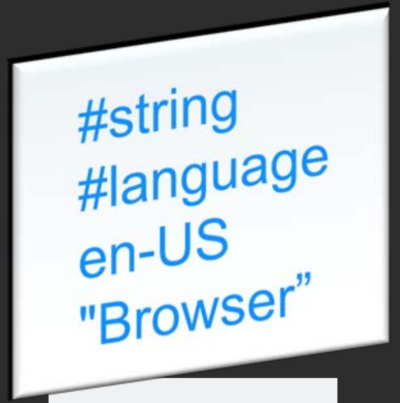
UEFI Driver Initialization Process

HII Protocols	
Config Routing Protocol	
	ExtractConfig
	RouteConfig
	ExportConfig
	BlockToConfig
	ConfigToBlock
Form Browser 2 Protocol	
	SendForm
	BrowserCallback
HII Database Protocols	
	NewPackageList
	Remove
	Update
	. . .
	GetPackageListHandle

MyDriver

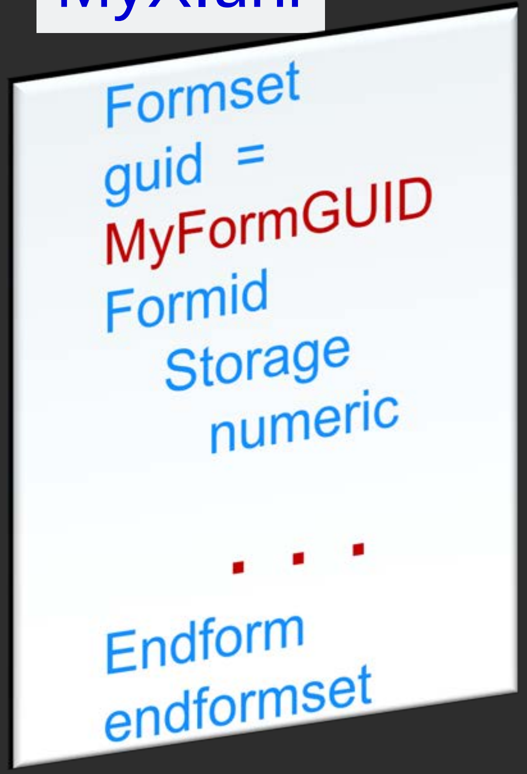
UEFI 2.x+ Driver (e.g. Motherboard Driver, Addin card Op ROM)
Config Access Protocol
ExtractConfig
RouteConfig
Call Back

1. Produce Config Access Protocols



#string
#language
en-US
"Browser"

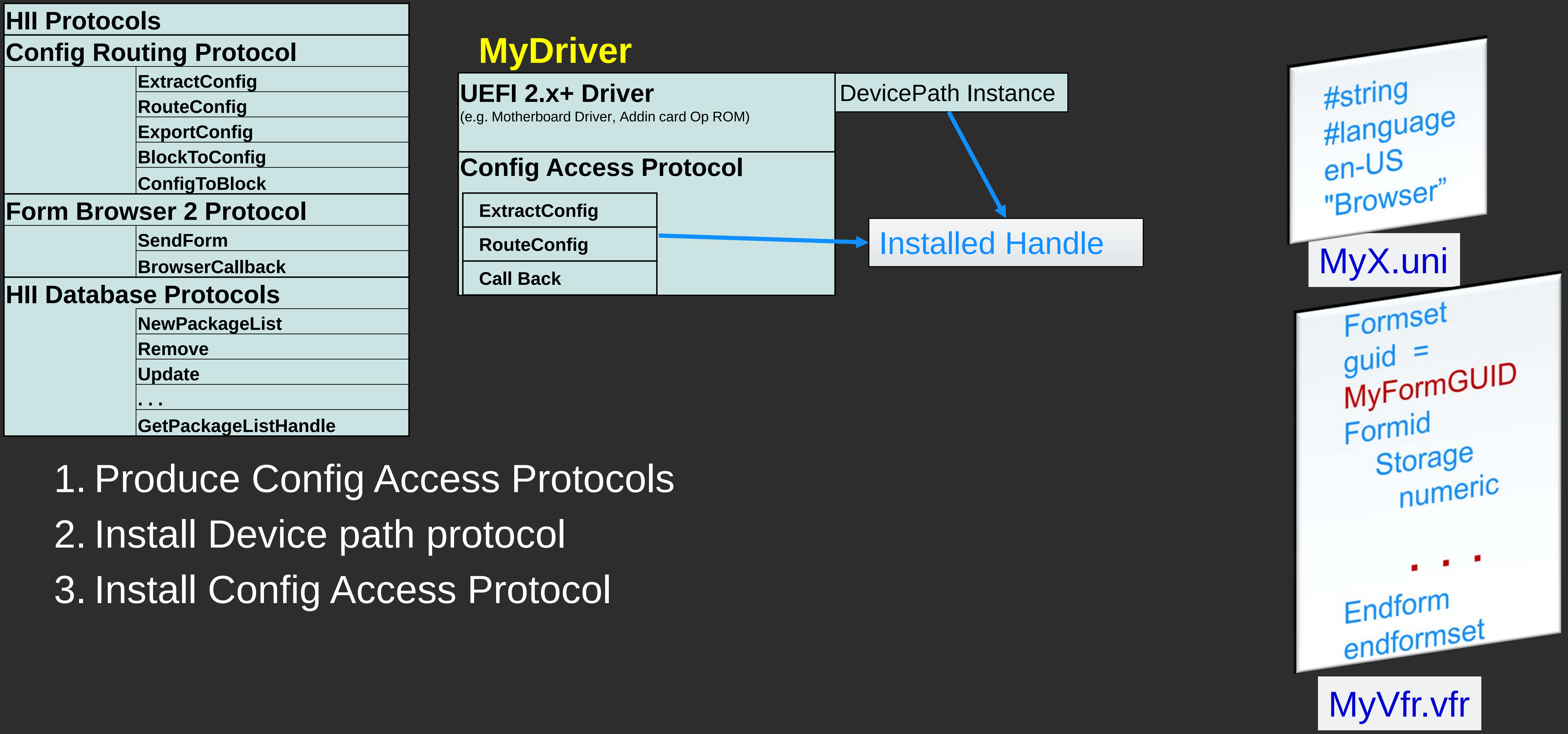
MyX.uni



Formset
guid = MyFormGUID
Formid
Storage
numeric
. . .
Endform
endformset

MyVfr.vfr

UEFI Driver Initialization Process



1. Produce Config Access Protocols
2. Install Device path protocol
3. Install Config Access Protocol

tianocore

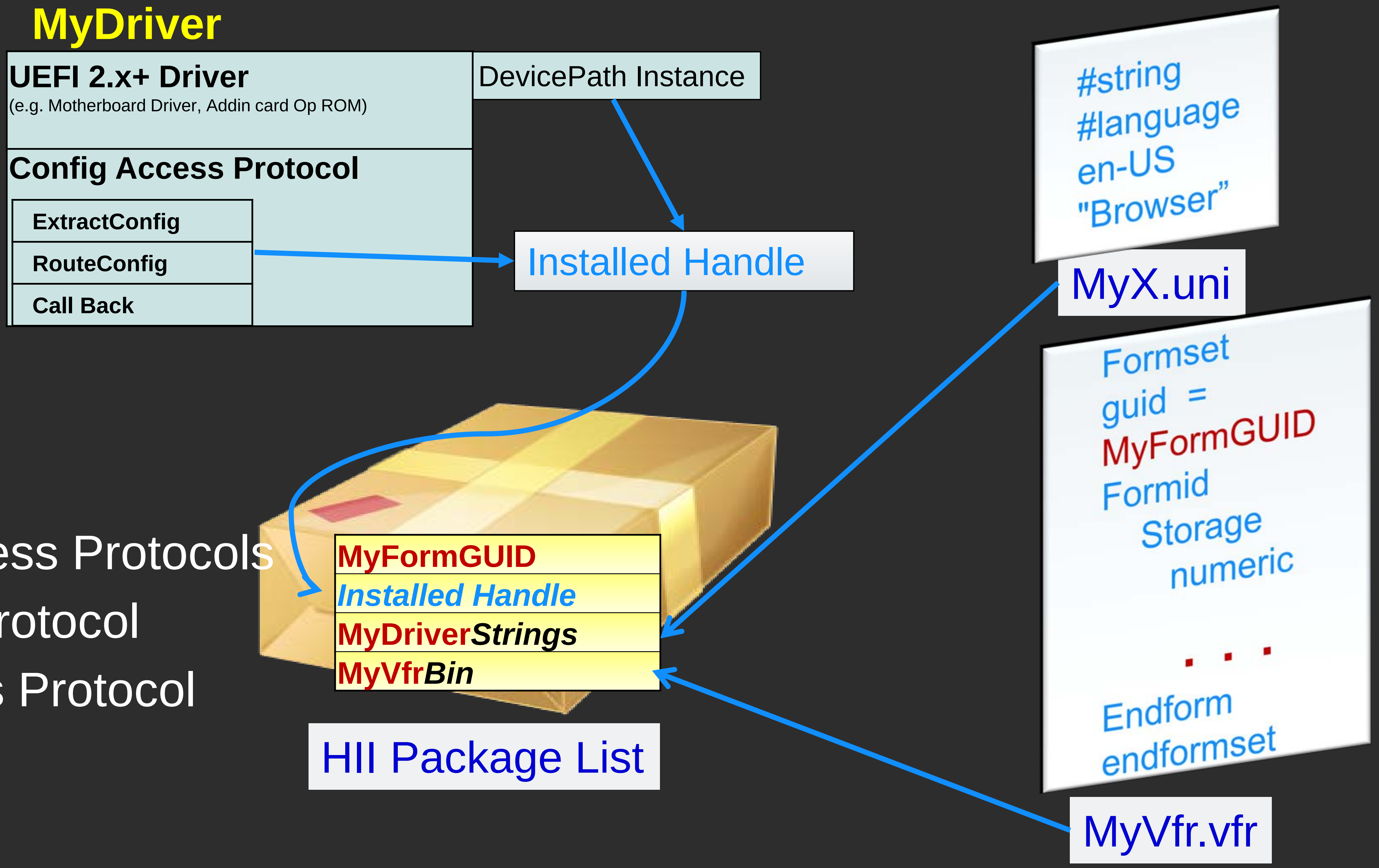
UEFI Driver Initialization Process

HII Protocols

Config Routing Protocol
- ExtractConfig
- RouteConfig
- ExportConfig
- BlockToConfig
- ConfigToBlock

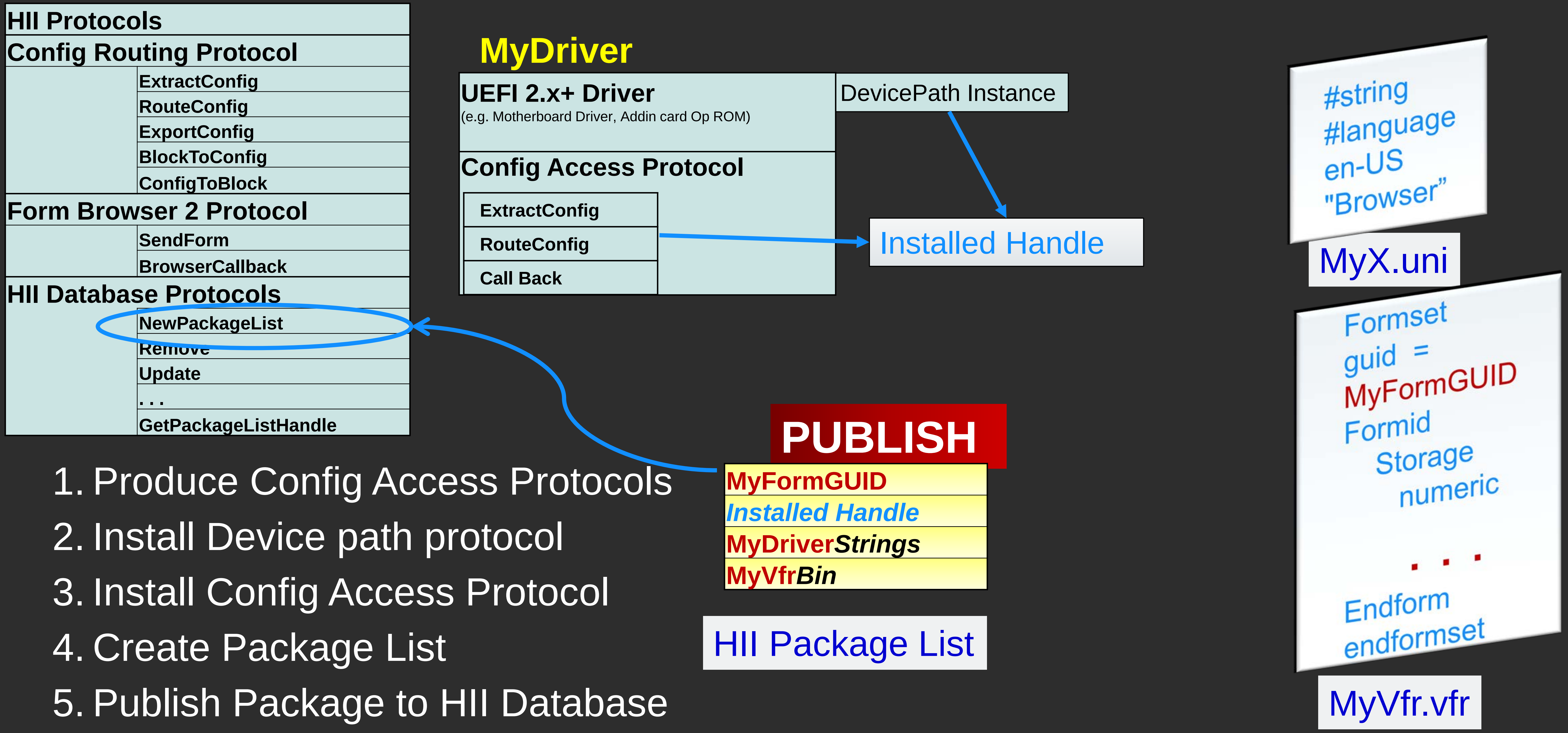
Form Browser 2 Protocol
- SendForm
- BrowserCallback

HII Database Protocols
- NewPackageList
- Remove
- Update
- . . .
- GetPackageListHandle



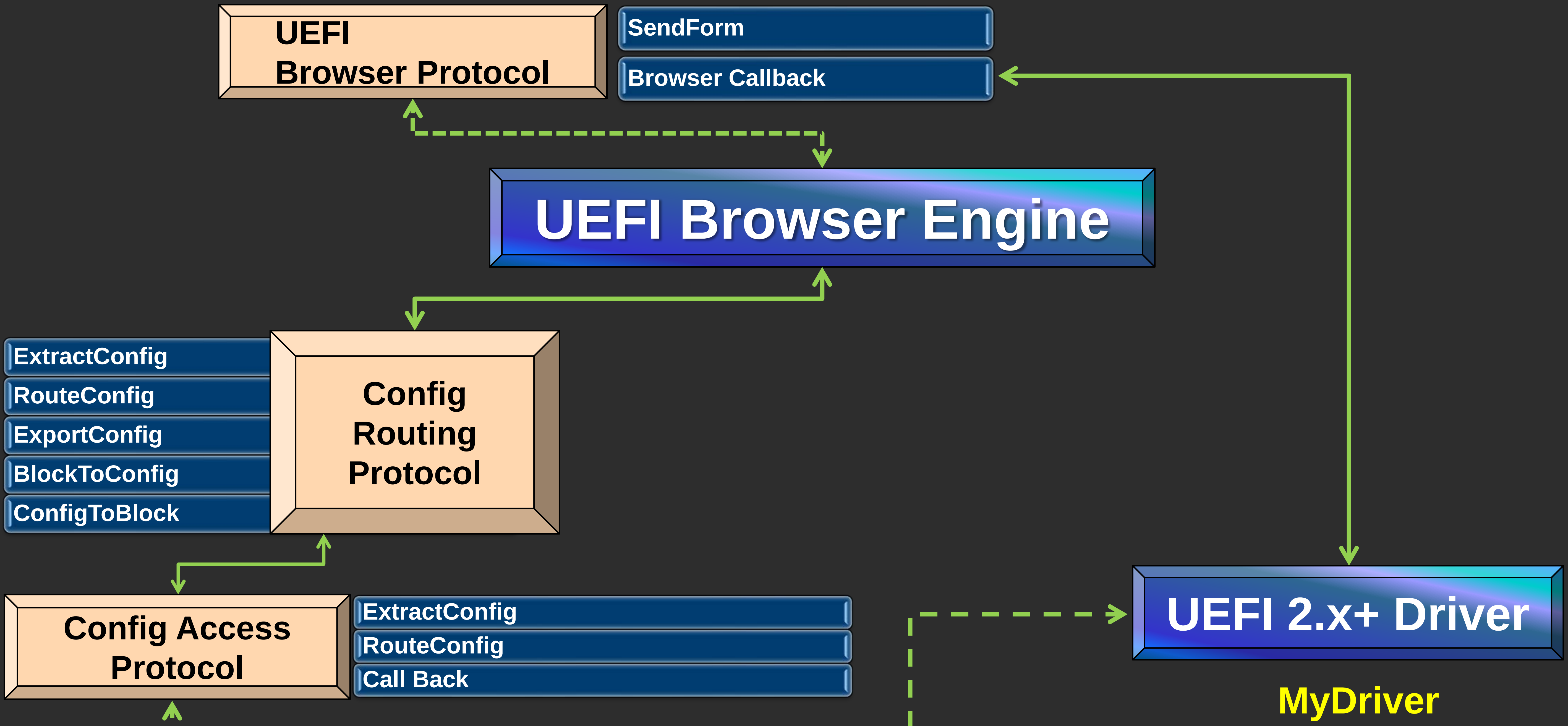
1. Produce Config Access Protocols
2. Install Device path protocol
3. Install Config Access Protocol
4. Create Package List

UEFI Driver Initialization Process



1. Produce Config Access Protocols
2. Install Device path protocol
3. Install Config Access Protocol
4. Create Package List
5. Publish Package to HII Database

Form Browser Protocols



tianocore

LAB FOR HII

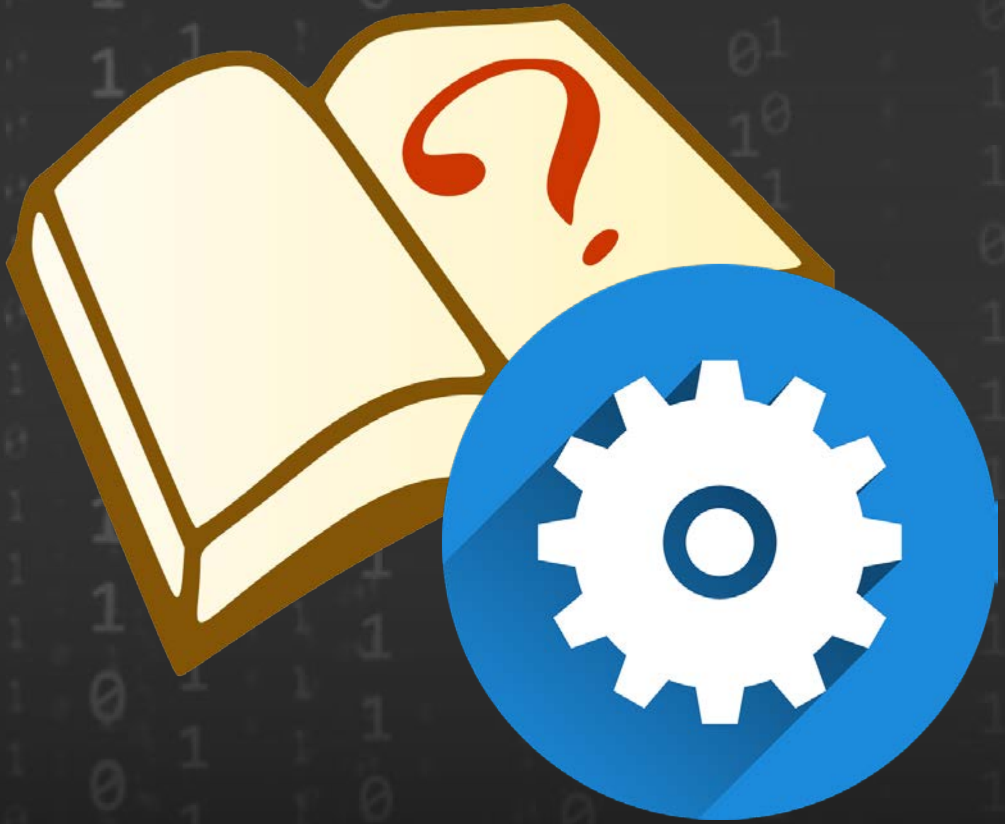
tianocore

LAB FOR HII

Use the Lab guide to follow the steps Adding HII to a UEFI Driver from the UEFI Driver Wizard Lab

- link to pdf Linux
- link to pdf Windows

Perquisite UEFI Driver Porting Lab



tianocore

Reference

Unified Extensible Firmware Interface Specification, Version 2.7, http://www.uefi.org (UEFI 2.1 or greater needed for HII)

VFR Programming Language 1.92, https://github.com/tianocore/tianocore.github.io/wiki/EDK-II-Specifications#vfr

Build Spec 1.28, https://github.com/tianocore/tianocore.github.io/wiki/EDK-II-Specifications#build

Summary

- What is the Infrastructure for HII
- How Does HII Work
- Lab for HII

tianocore

Questions?

Return to Main Training Page

Return to Training Table of contents for next presentation link

tianocore

tianocore

www.tianocore.org

tianocore

ACKNOWLEDGEMENTS