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

tianocore

USER INTERFACE HII OVERVIEW

tianocore

WHY ?

Unified Look and Feel at Platform level

Single Interface

Localization

• • •

HII: KEY CONCEPTS

forms & strings

HII: KEY CONCEPTS

forms & strings

HII

HII: KEY CONCEPTS

forms & strings

HII

HII: KEY CONCEPTS

forms & strings

HII

localization

HII: KEY CONCEPTS

forms & strings

HII

localization

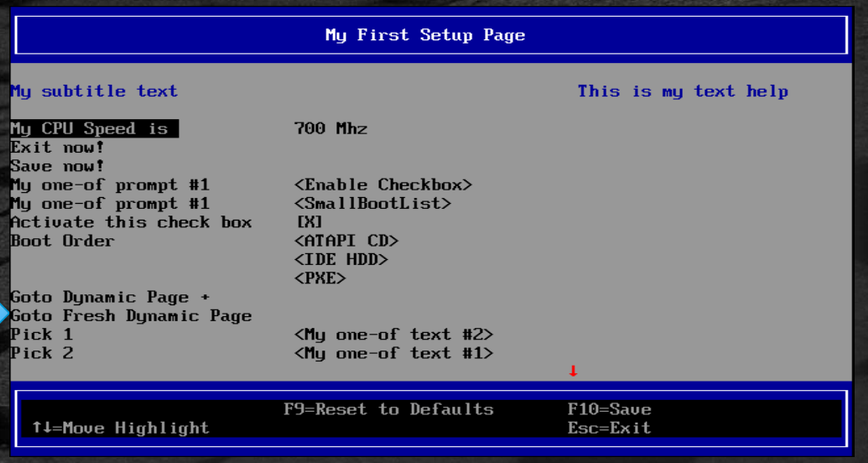
input sources

HII: KEY CONCEPTS

forms & strings

HII

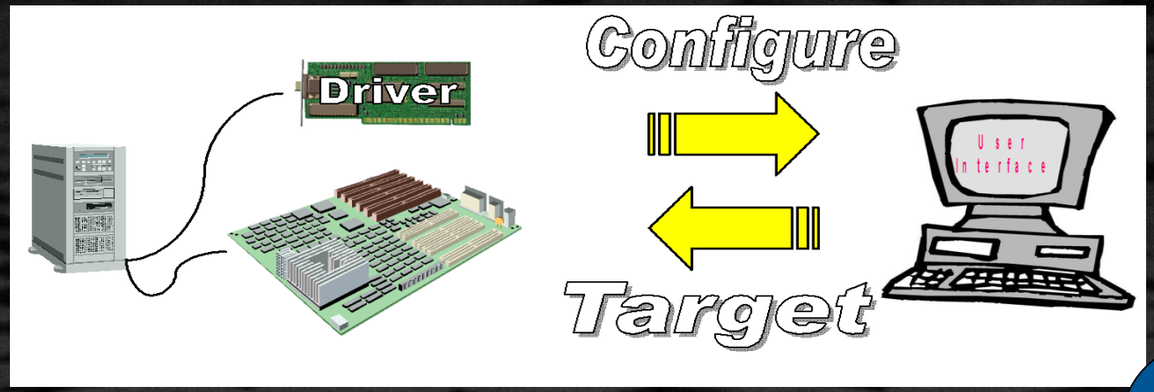
localization

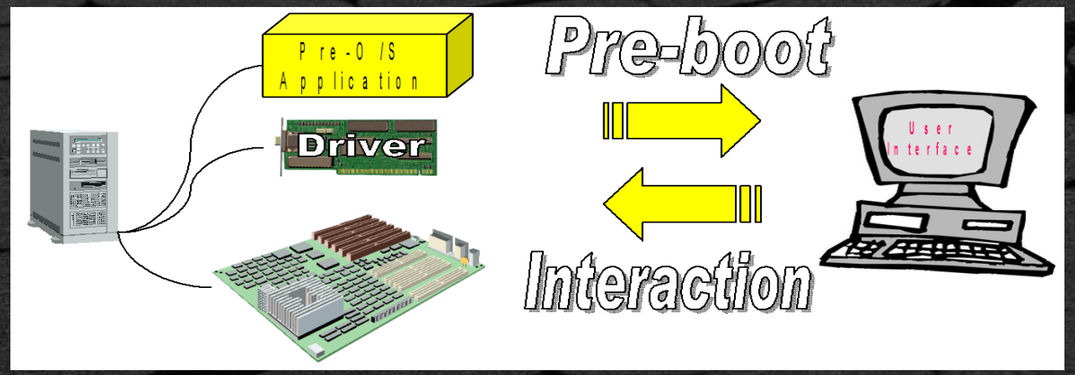


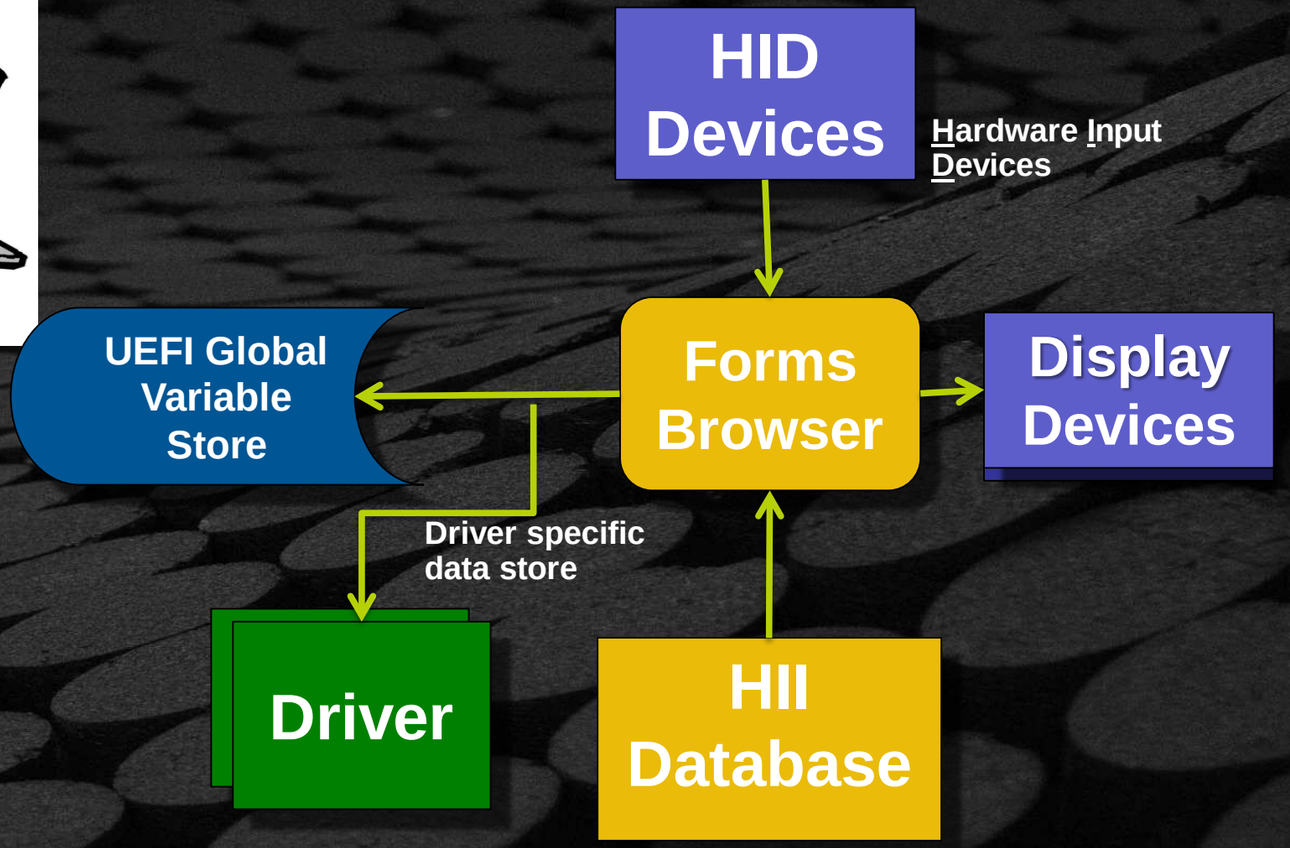
setup browser

input sources

DESIGN DISCUSSIONS







See § 29.2 of the UEFI 2.x Spec.

tianocore

HII COMPONENTS

tianocore

HUMAN INTERFACE COMPONENTS

Strings

TEXT

tianocore

HUMAN INTERFACE COMPONENTS

Strings

TEXT

Fonts

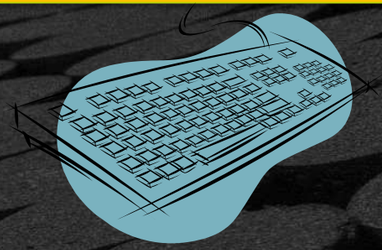
HUMAN INTERFACE COMPONENTS

Strings

TEXT

Fonts

Keyboard



tianocore

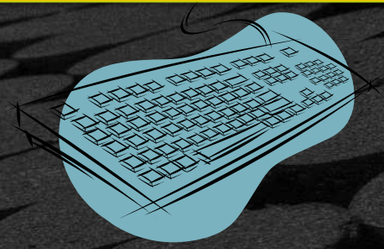
HUMAN INTERFACE COMPONENTS

Strings

TEXT

Fonts

Keyboard



Forms

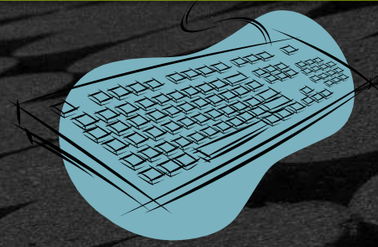
tianocore

HUMAN INTERFACE COMPONENTS

Strings

Fonts

Keyboard



Forms

Packages

STRINGS

Strings stored in Unicode

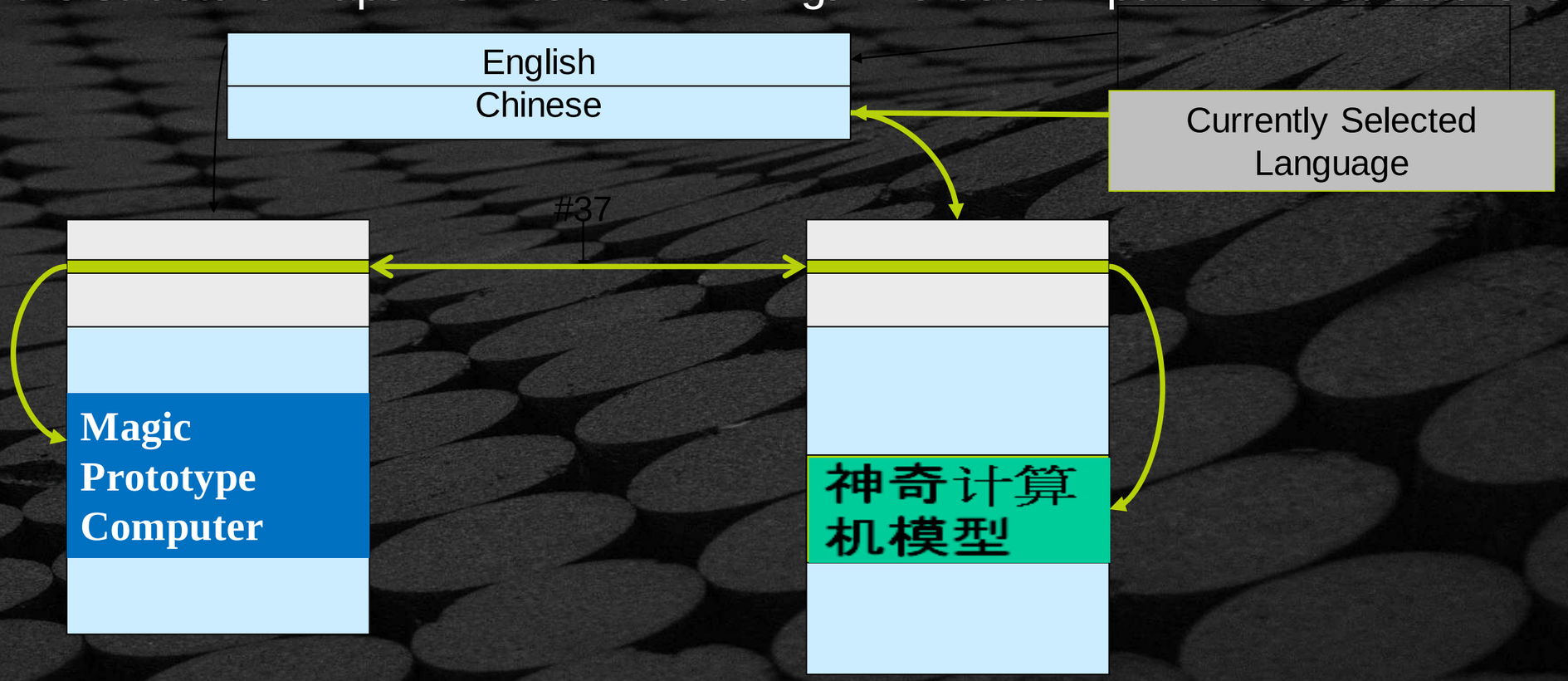
- Real string encodings required for e.g. VT100
- Already the text standard in UEFI today

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)

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



STRING EXAMPLE (.UNI FILE)


```
#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

tianocore

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

tianocore

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

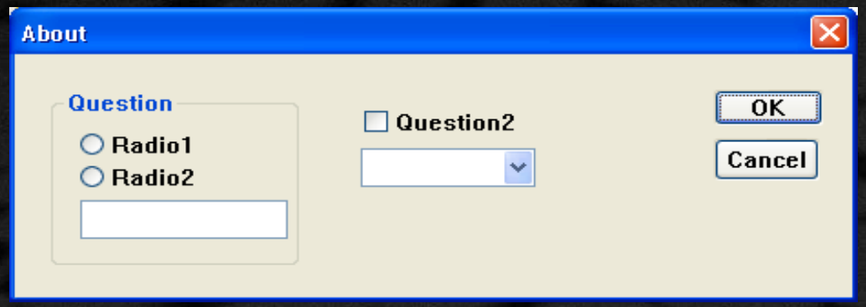
- Allow for a standardized mechanism to describe a keyboard layout and add to system database.
- Allow for switching keyboard layouts.

Spanish

English

French

tianocore



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.

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

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 …

MINIMUM FILES FOR HII DRIVER

.c

.h

MINIMUM FILES FOR HII DRIVER

.c

.h

.uni

.vfr

MINIMUM FILES FOR HII DRIVER

.c

.h

.uni

.vfr

.inf

tianocore

MINIMUM FILES FOR HII DRIVER

.c

.h

.uni

Strings

.vfr

Forms

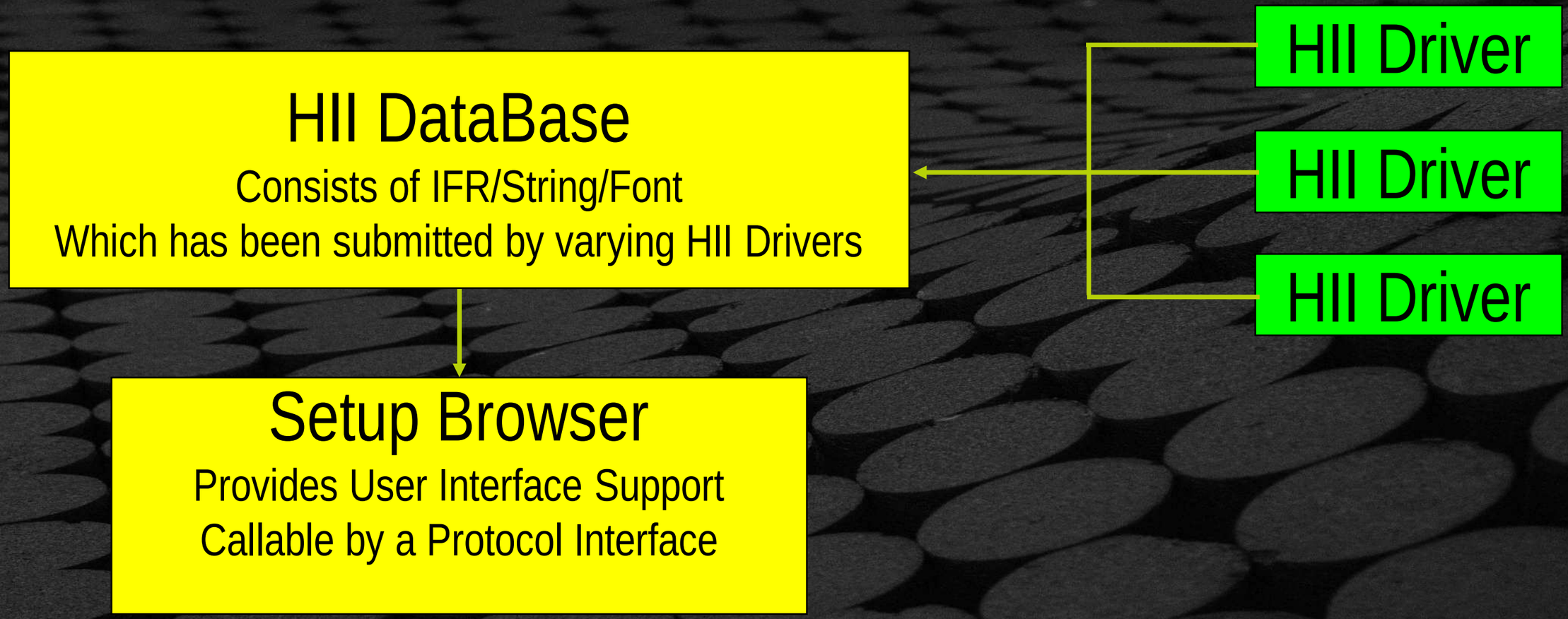
.inf

EDK II HII

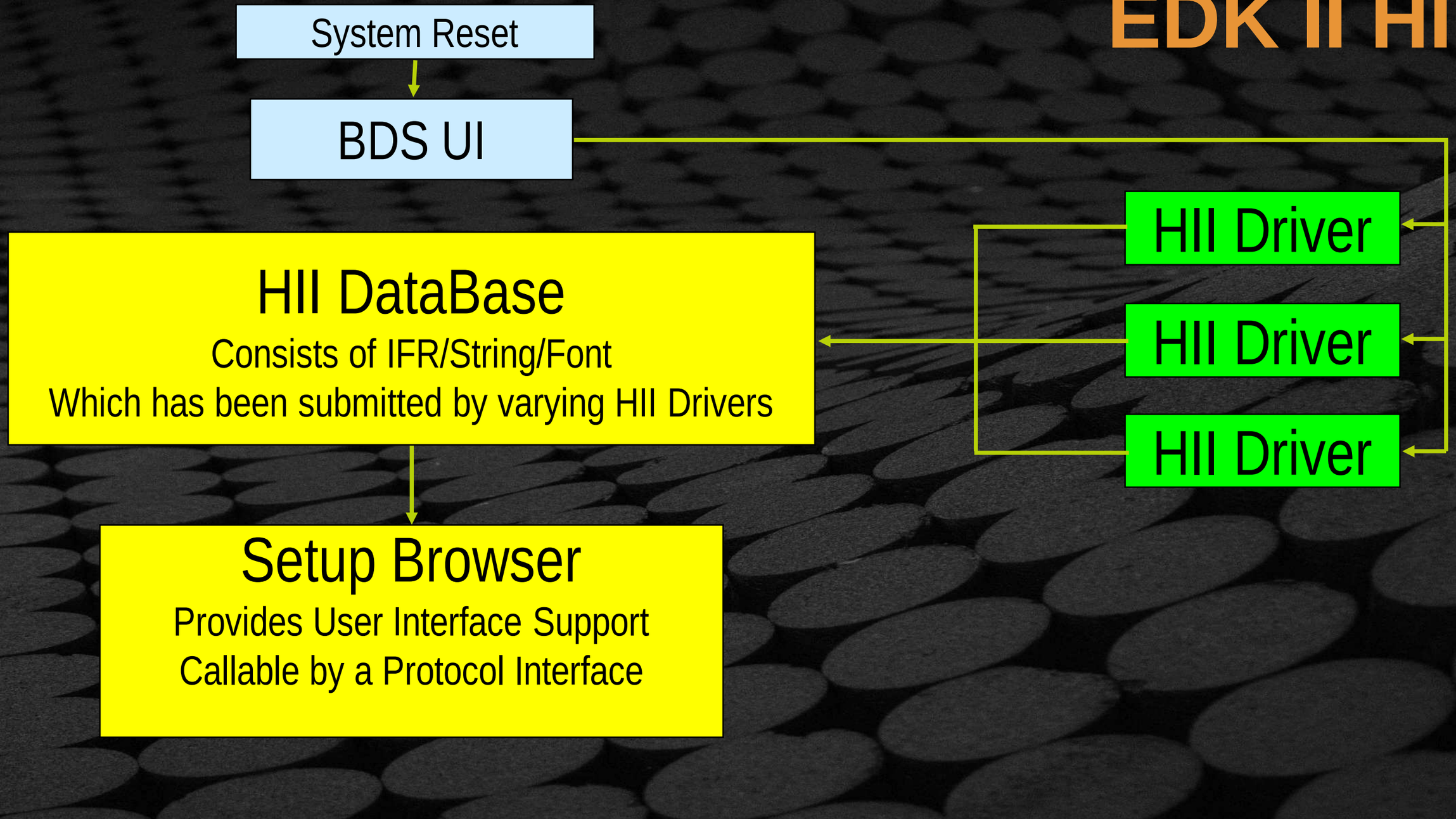

HII DataBase
Consists of IFR/String/Font
Which has been submitted by varying HII Drivers
HII Driver
HII Driver
HII Driver

tianocore

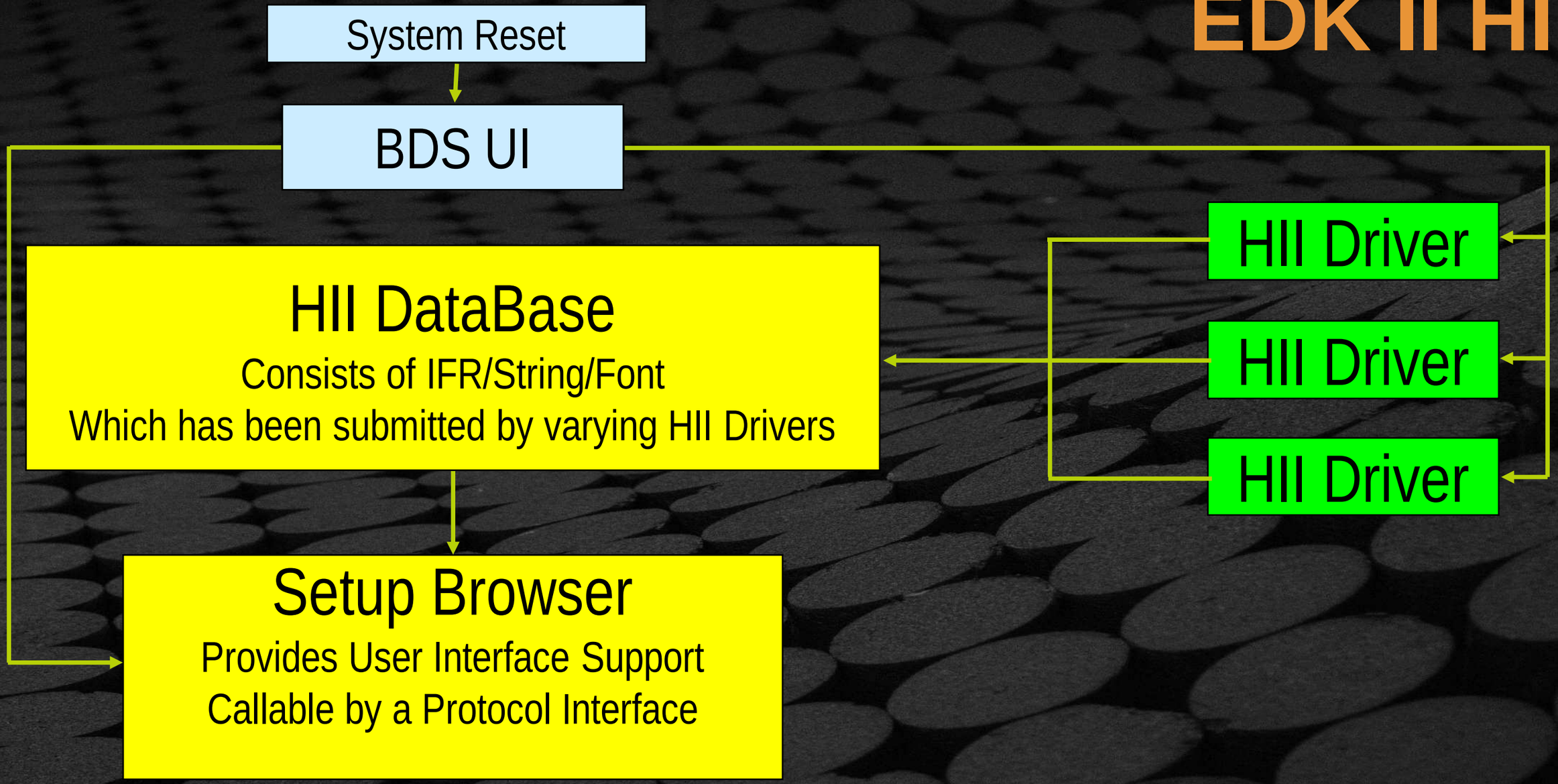
EDK II HII



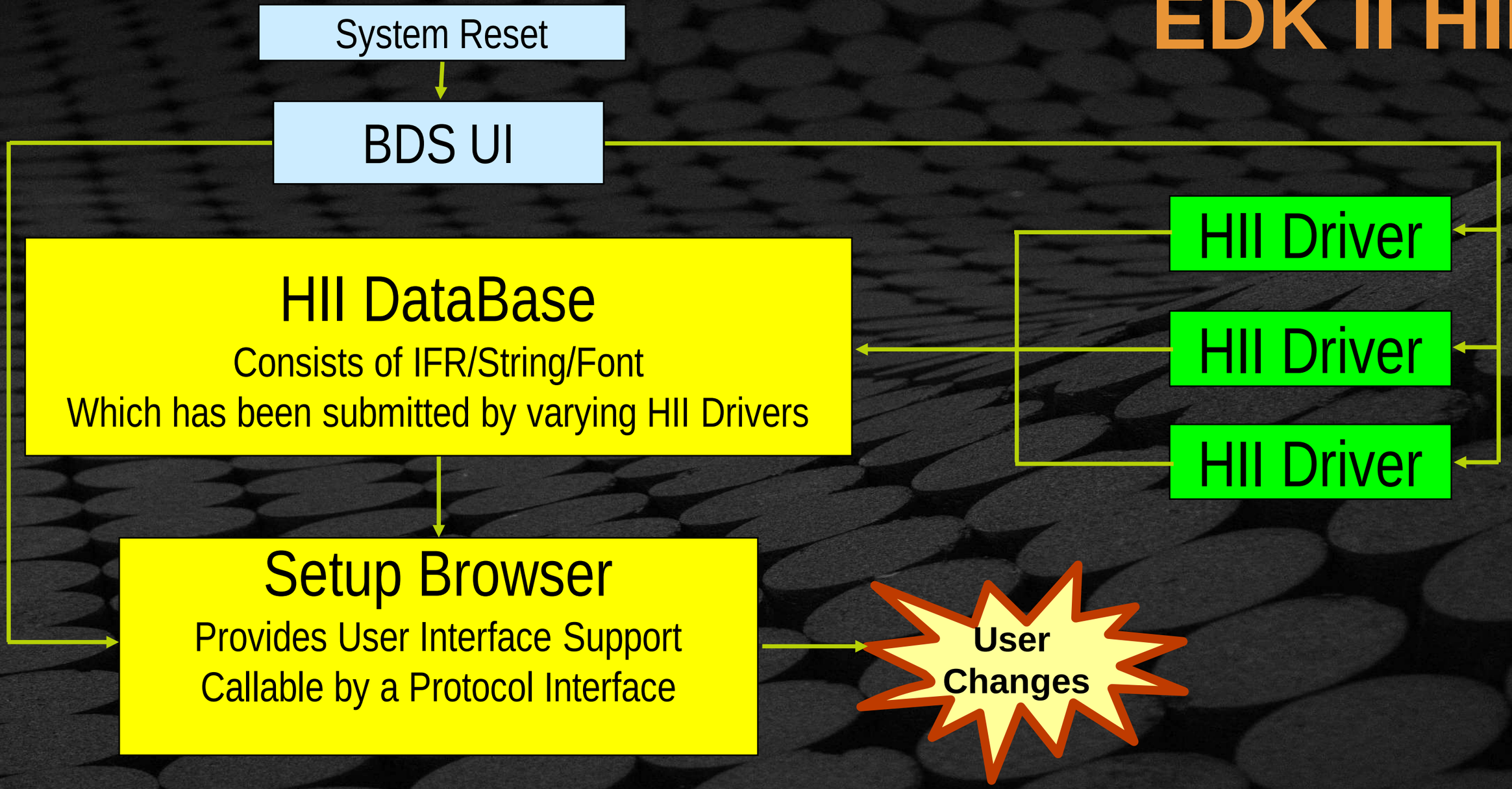
EDK II HII


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

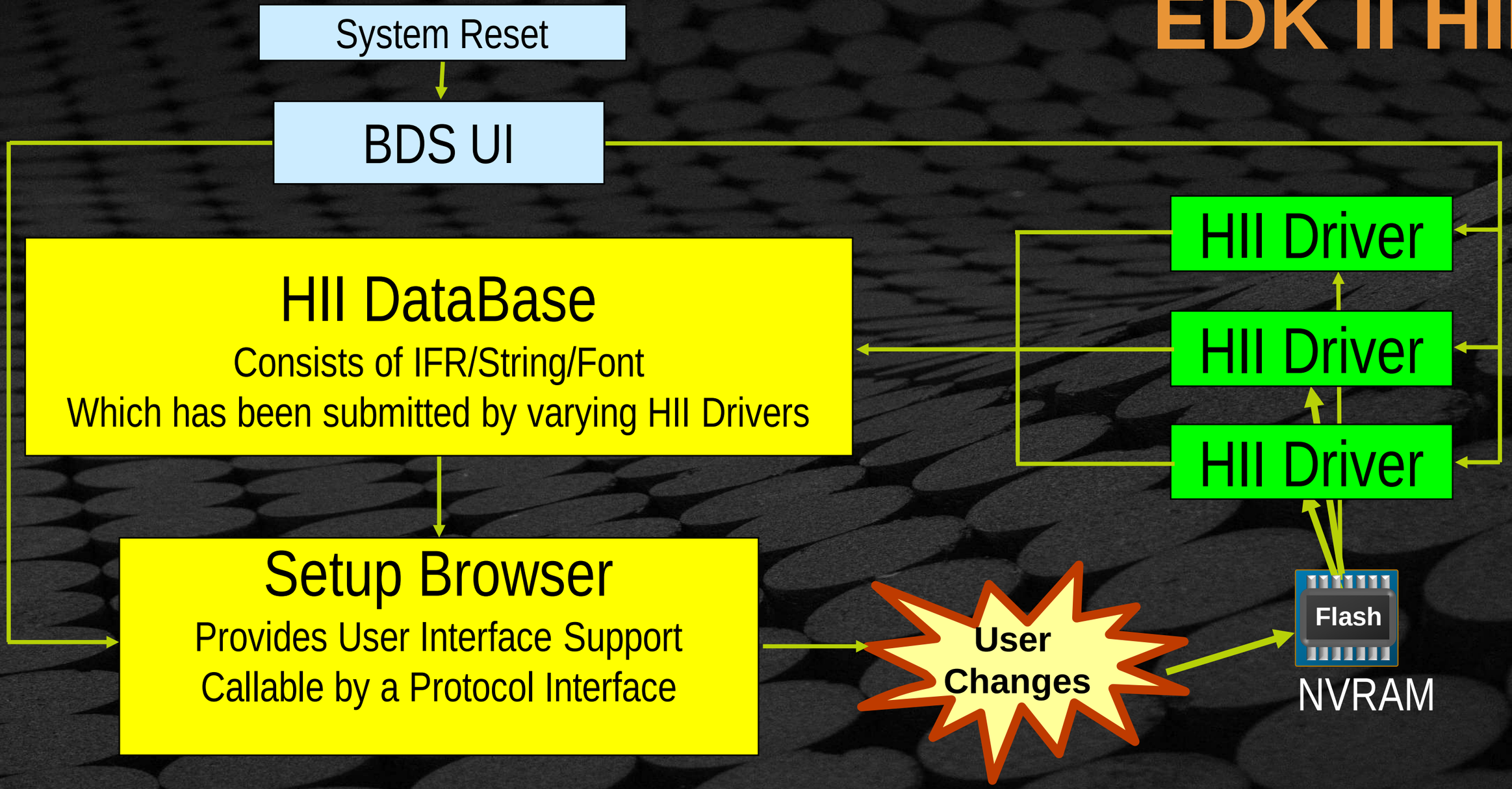

System Reset
BDS UI
HII DataBase
Consists of IFR/String/Font
Which has been submitted by varying HII Drivers
HII Driver
HII Driver
HII Driver
Setup Browser
Provides User Interface Support
Callable by a Protocol Interface

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

tianocore

EDK II HII



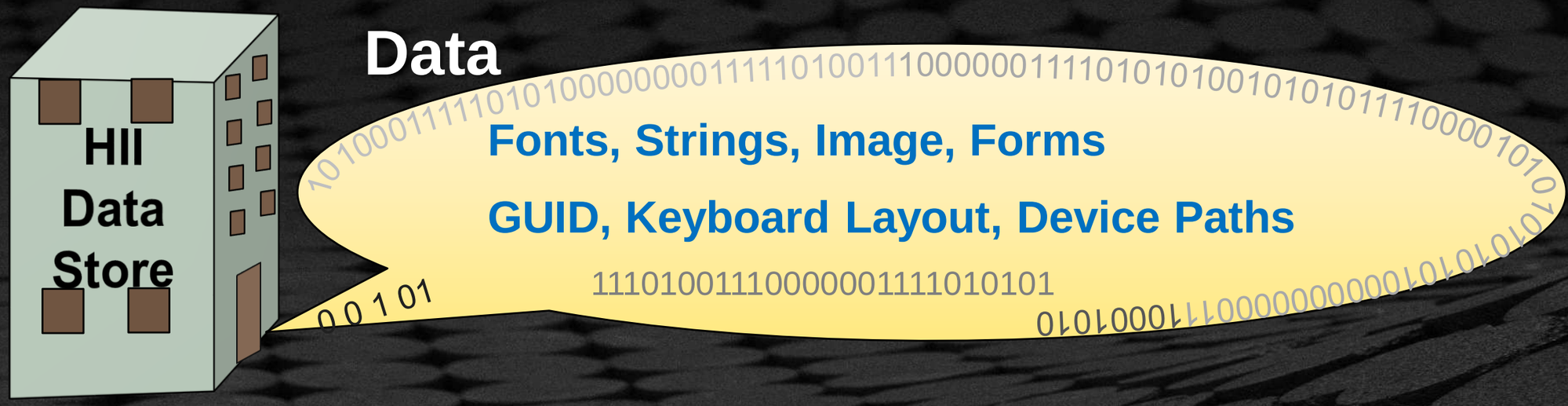
tianocore

HOW: UEFI HII PROTOCOLS

Sections 29-31 the UEFI 2.x Specification

HII DATABASE OVERVIEW

HII DATABASE OVERVIEW



tianocore

HII DATABASE OVERVIEW

Data

Fonts, Strings, Image, Forms

GUID, Keyboard Layout, Device Paths

0 0 1 01

01 0 1 01

01 0 1 01

01 0 10001

HII Protocols

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

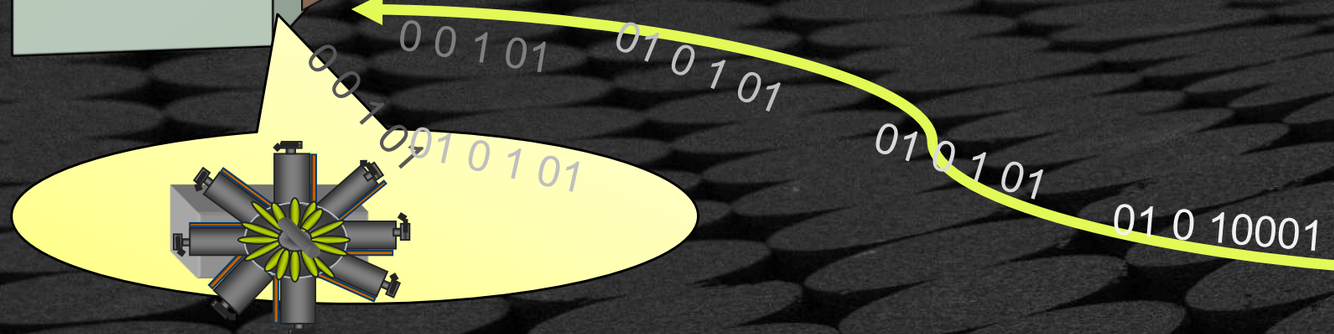
See § 30 of the UEFI 2.x Spec.

HII DATABASE OVERVIEW

Data

Fonts, Strings, Image, Forms

GUID, Keyboard Layout, Device Paths



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.

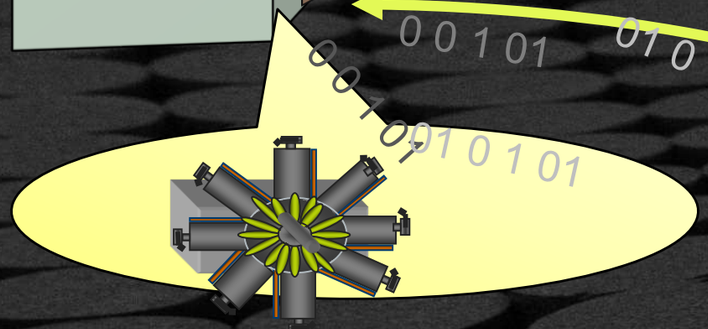
HII DATABASE OVERVIEW

Data

Fonts, Strings, Image, Forms

GUID, Keyboard Layout, Device Paths

NVRAM



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.

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

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"
```

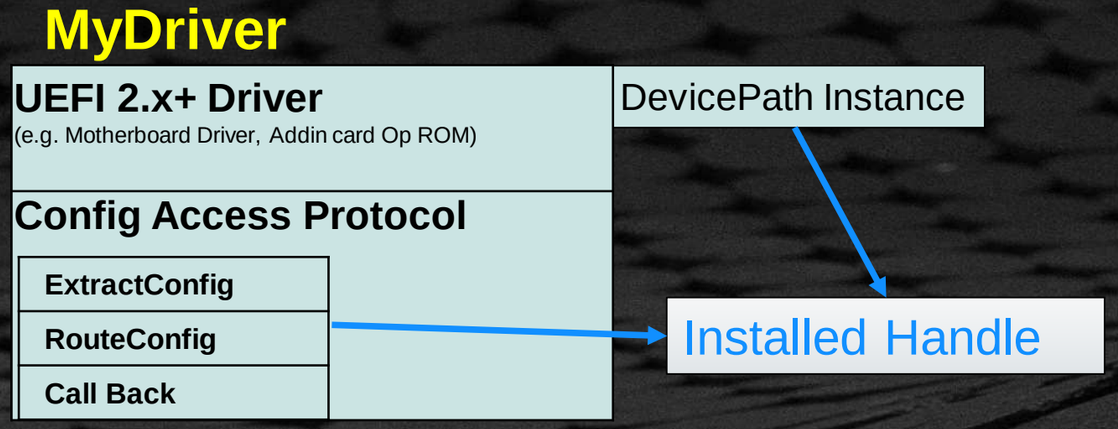
MyX.uni

```
Formset
guid =
MyFormGUID
Formid
  Storage
    numeric

    . . .

Endform
endformset
```

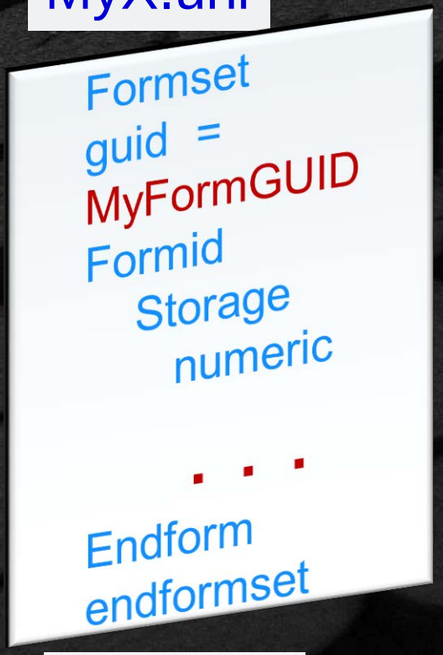
MyVfr.vfr

UEFI DRIVER INITIALIZATION PROCESS

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



MyX.uni



MyVfr.vfr

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

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)

DevicePath Instance

Config Access Protocol

- ExtractConfig
- RouteConfig
- Call Back

Installed Handle

#string
#language
en-US
"Browser"

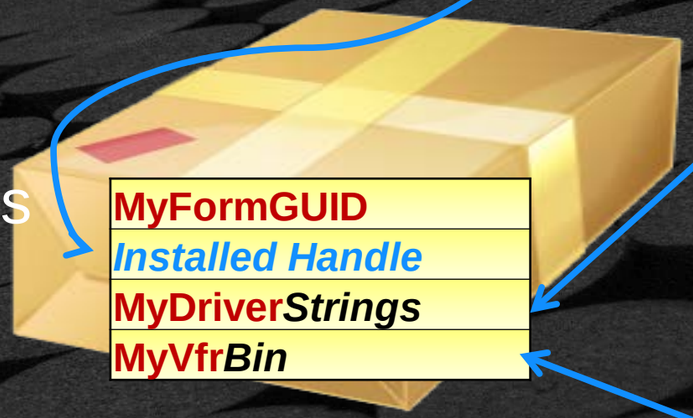
MyX.uni

Formset
guid =
MyFormGUID
Formid
Storage
numeric

. . .

Endform
endformset

MyVfr.vfr



HII Package List

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

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)

DevicePath Instance

Config Access Protocol

- ExtractConfig
- RouteConfig
- Call Back

Installed Handle

#string
#language
en-US
"Browser"

MyX.uni

Formset
guid =
MyFormGUID
Formid
Storage
numeric

. . .

Endform
endformset

MyVfr.vfr

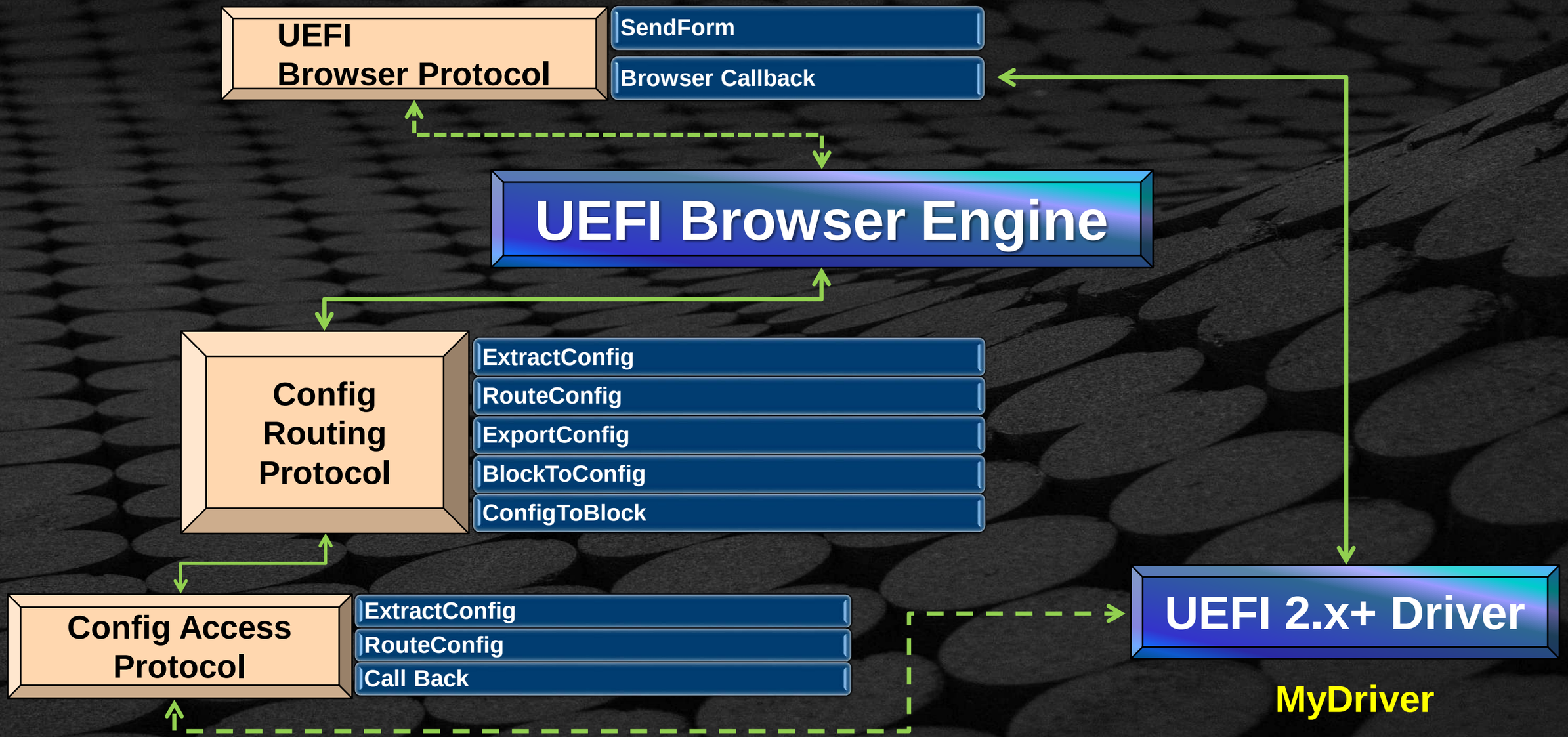
PUBLISH

MyFormGUID
Installed Handle
MyDriver*Strings*
MyVfr*Bin*

HII Package List

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

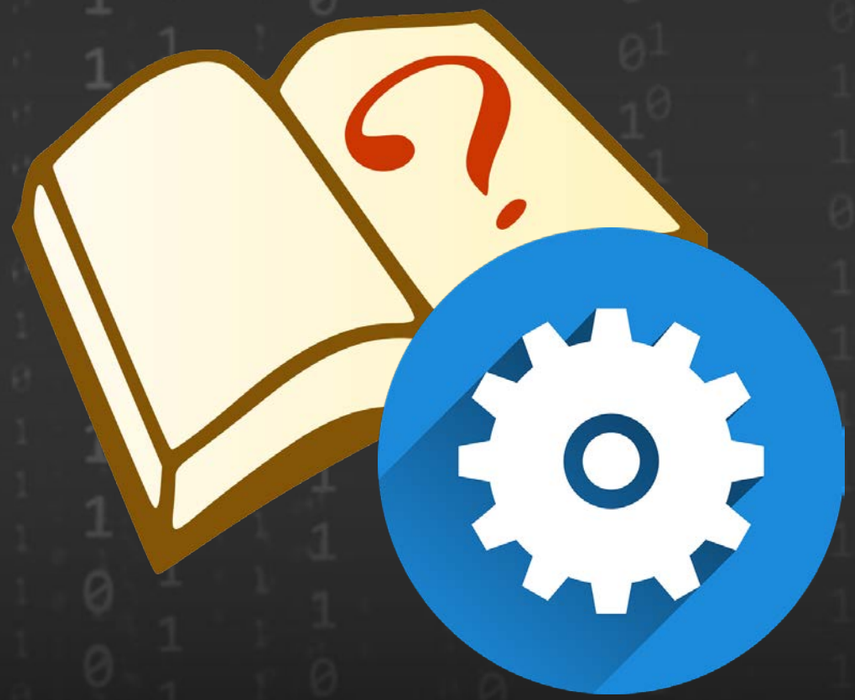
LAB FOR HII

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



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

tianocore

LESSON OBJECTIVE

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

tianocore

Questions?

tianocore