Software Frameworks

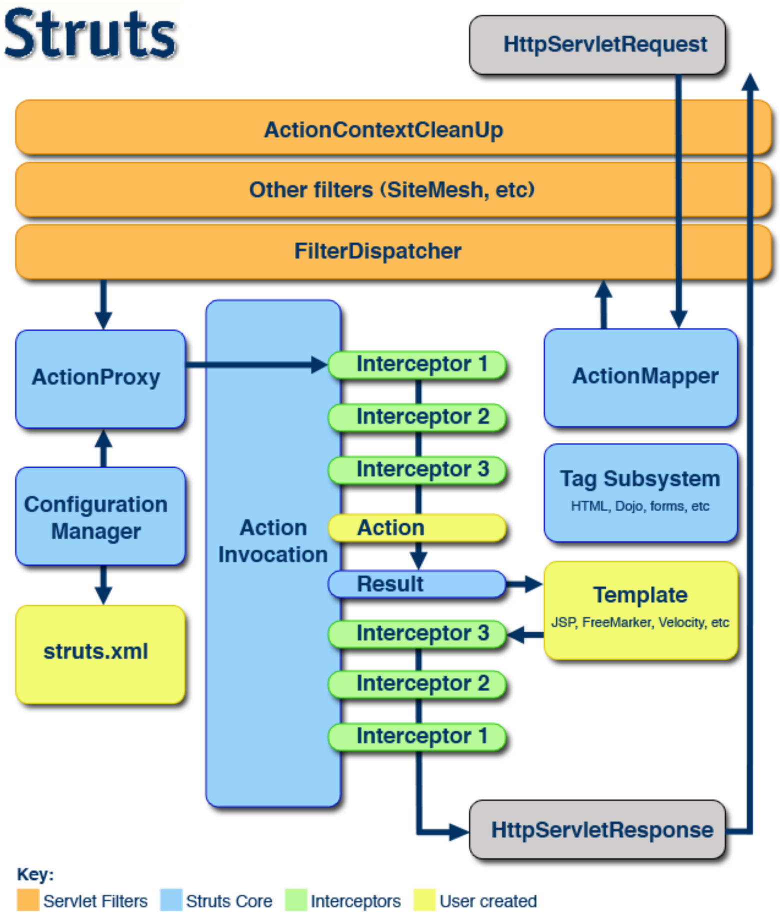
CHANDAN R. RUPAKHETI

WEEK 1-3

Software Frameworks

"A software framework **is an abstraction** in which software providing **generic functionality** can be **selectively changed by additional user-written code** to provide application-specific software."

- Wikipedia



Framework vs. Library 1/2

Inversion of Control

- Overall program flow is dictated by framework not by caller

Default Behavior

- A framework has a useful default behavior

Framework vs. Library 2/2

Extensibility

- A framework can be extended by the user

Non-Modifiability

- Users can extend the framework, but not modify its code

The Good, the Bad, and the Ugly 1/2

Frameworks helps developer **focus on domain-logic** rather than lower-level plumbing details

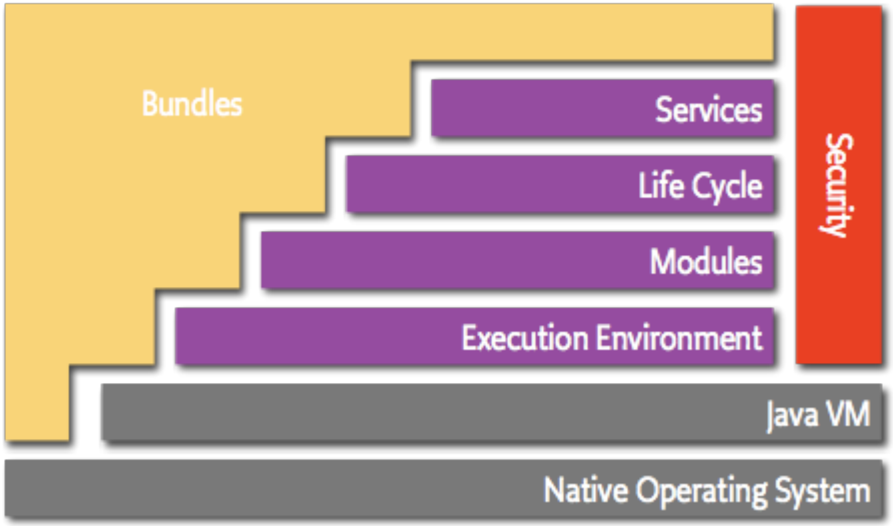
Frameworks have **longer learning curve, but productivity increases for similar kind of solutions**

The Good, the Bad, and the Ugly 2/2

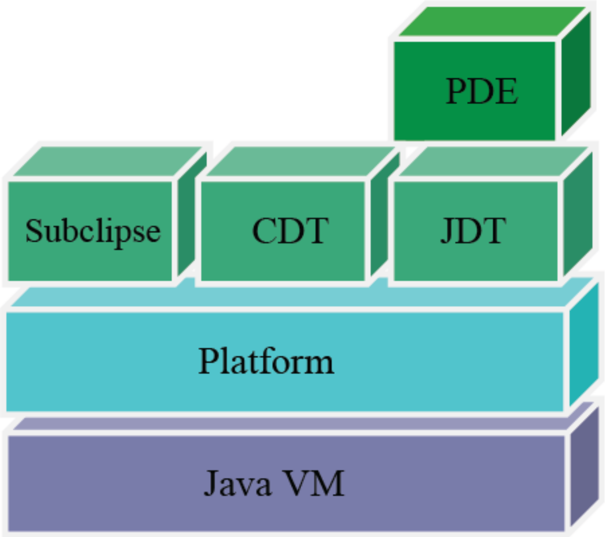
Frameworks often add to the program-size, which is also known as the **“code bloat”** phenomenon

Software elegance implies **clarity, conciseness, and little waste, but frameworks cannot always guarantee** these properties

Pluggable Software Architecture



OSGi Architecture



The Eclipse Software Stack

Definition 1/2

What is a plug-in?

"A small piece of software that supplements a larger program"

-Merriam-Webster's Online Dictionary

Definition 2/2

Some differentiate between an extension and a plug-in

Plug-ins

- Rely on user interface
- Are limited in their set of actions

Extensions

- Can add new functionality and user interface elements

Motivation

Keep software simple for end user

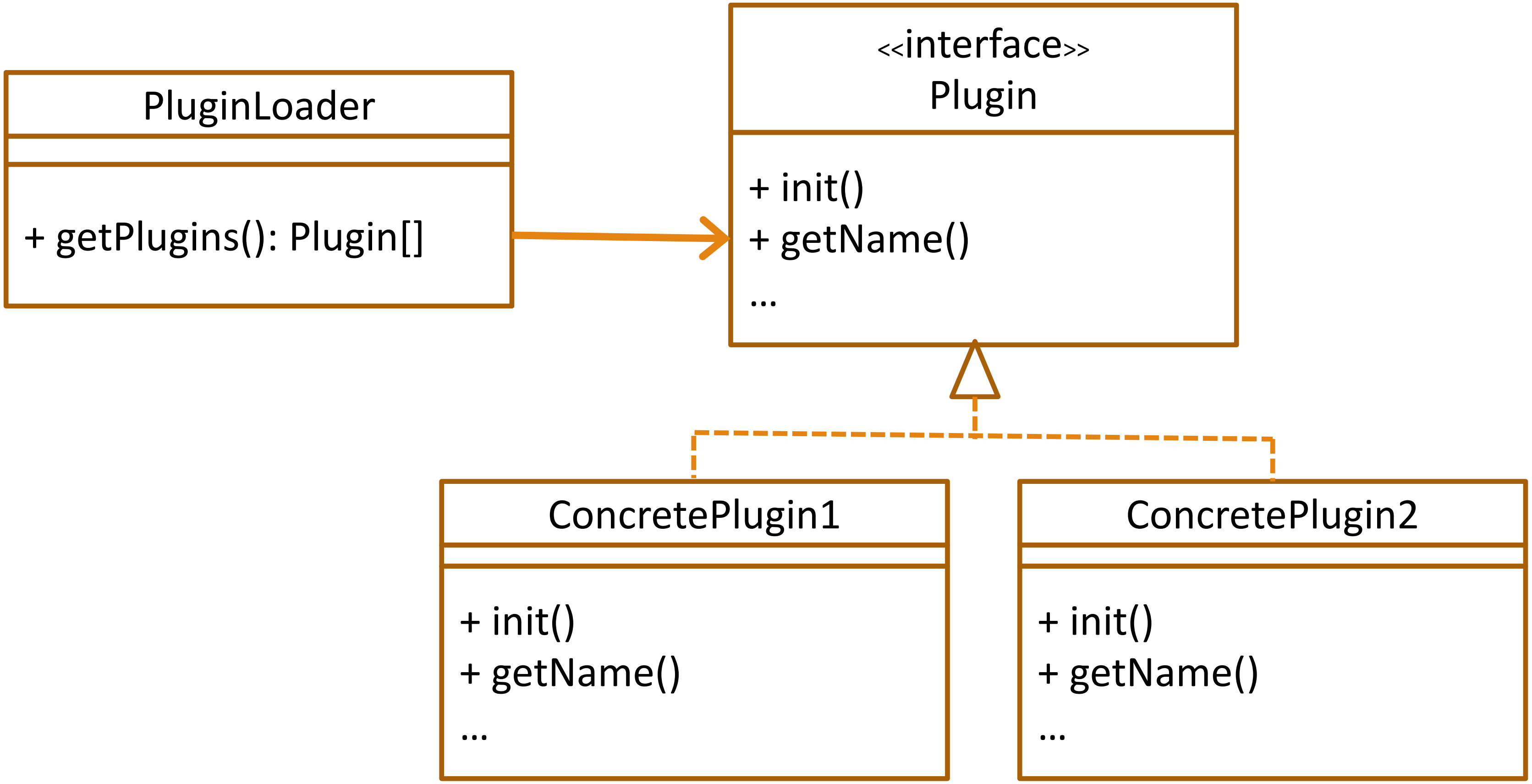
Modularity

Customizability

Extensibility

Resourcefulness

Plug-in Design



PluginLoader

+ getPlugins(): Plugin[]

<<interface>>
Plugin

+ init()
+ getName()
…

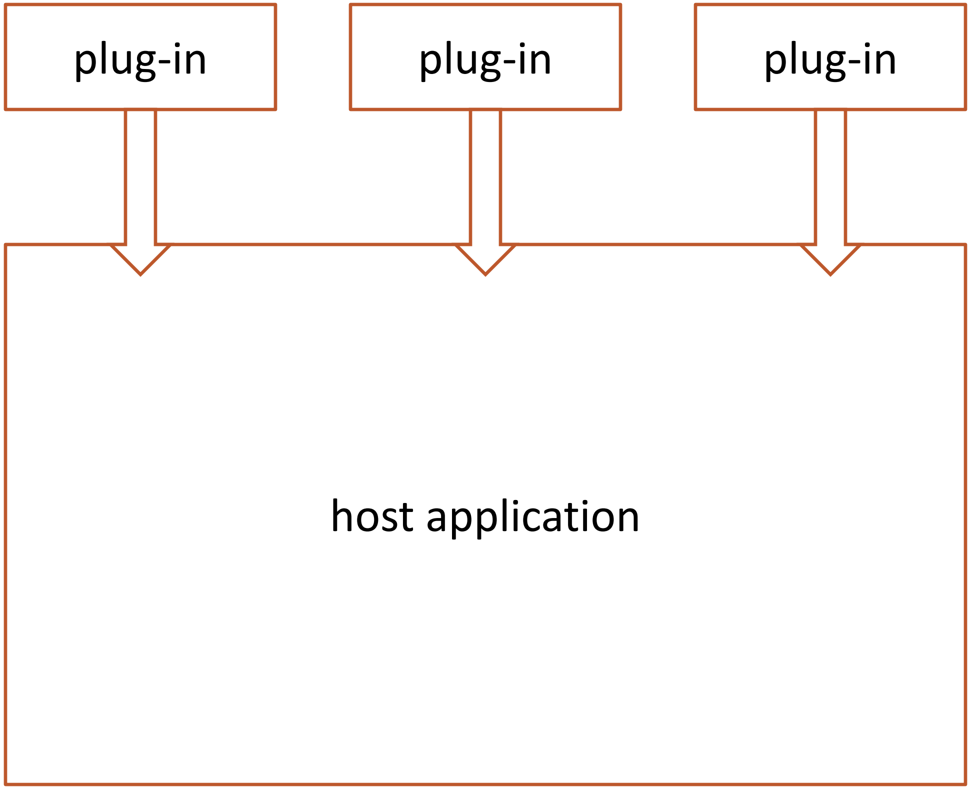
ConcretePlugin1

+ init()
+ getName()
…

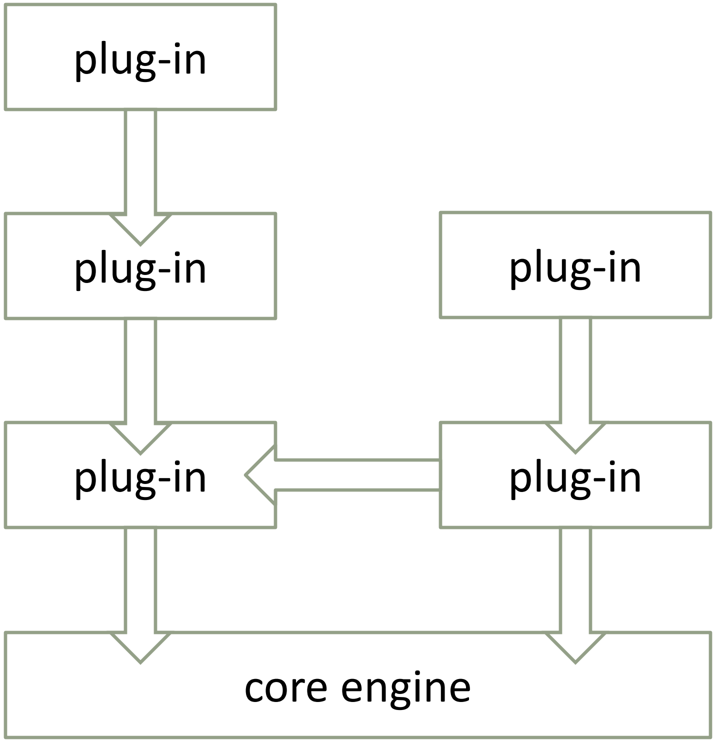
ConcretePlugin2

+ init()
+ getName()
…

Traditional Architecture



Pure Plug-in Architecture



Challenges

Discovery

Update

Security

Versions and Dependencies

Scalability

Easy development of new plug-ins

GUI changes

Photoshop

Fixed set of plug-in types:

- Image filter, Import, Export, Parser, Extension, Format, Color picker, Selection, Automation

Scans for 'PiPL' (plug-in property list)

- Description of what the plug-in is and its preferences

Plug-ins have defined entry point (“main”)

- Called by Photoshop when needed
- Call includes a 200-field structure

Q5

Firefox (Mozilla Products) 1/2

Distinction: Plug-ins and Extensions

Plug-ins (Gecko Plugin API)

- Help browser display data (pdf, multimedia, etc.)
- Register for MIME-types
- Draw into browser, modify DOM
- Receive mouse and keyboard events

Source: https://developer.mozilla.org/en-US/docs/Plugins

Firefox (Mozilla Products)   2/2

Extensions

- Modify browser’s GUI (written in XUL and Javascript)
  - XUL defines how the interface looks (static description)
  - Javascript defines how it works (interaction description)
- Can attach more elements to these at run time
- Can also modify behavior

Source: https://developer.mozilla.org/en-US/docs/Extensions

Microsoft Office

Plug-ins discovered by registry entries

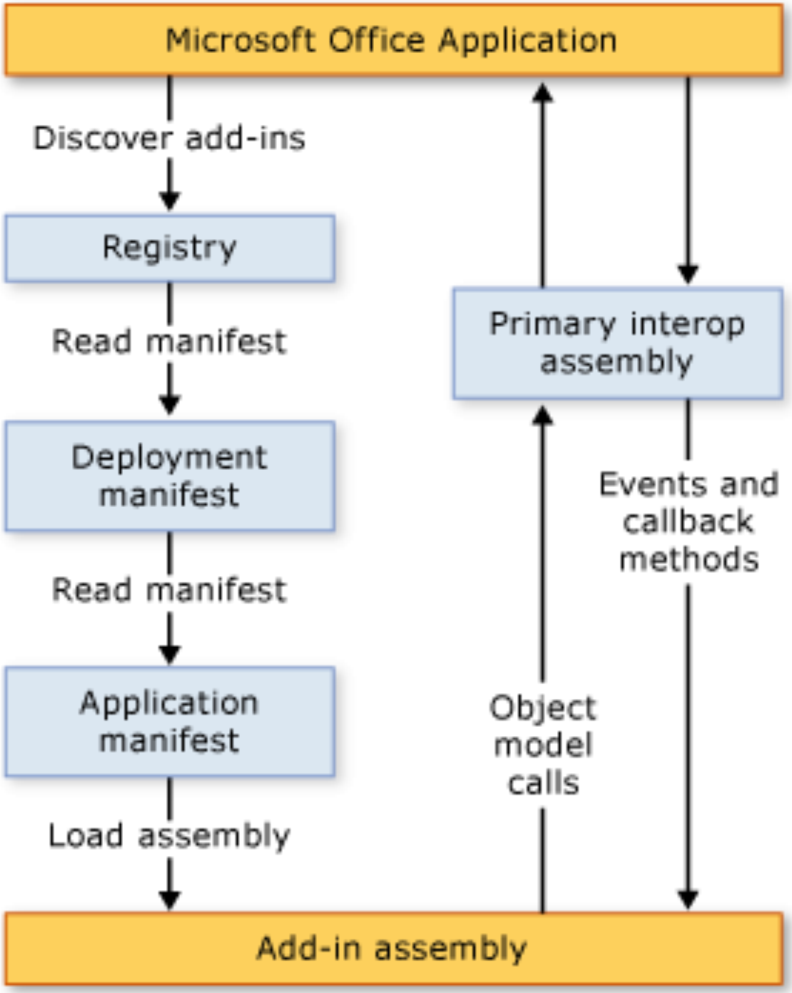
Can add things to the user interface

Have access to the Object Model

Can register callbacks for all kinds of events

Development inside of Visual Studio with VSTO

Microsoft Office



Source:
http://msdn.microsoft.com/en-us/library/bb386298.aspx

Eclipse 1/2

Eclipse is good for building IDEs

- But can do almost any application (including non-graphical)

Searches for plug-ins in a specific folder

Load on demand

Eclipse 2/2

Every plug-in has a description (manifest) file

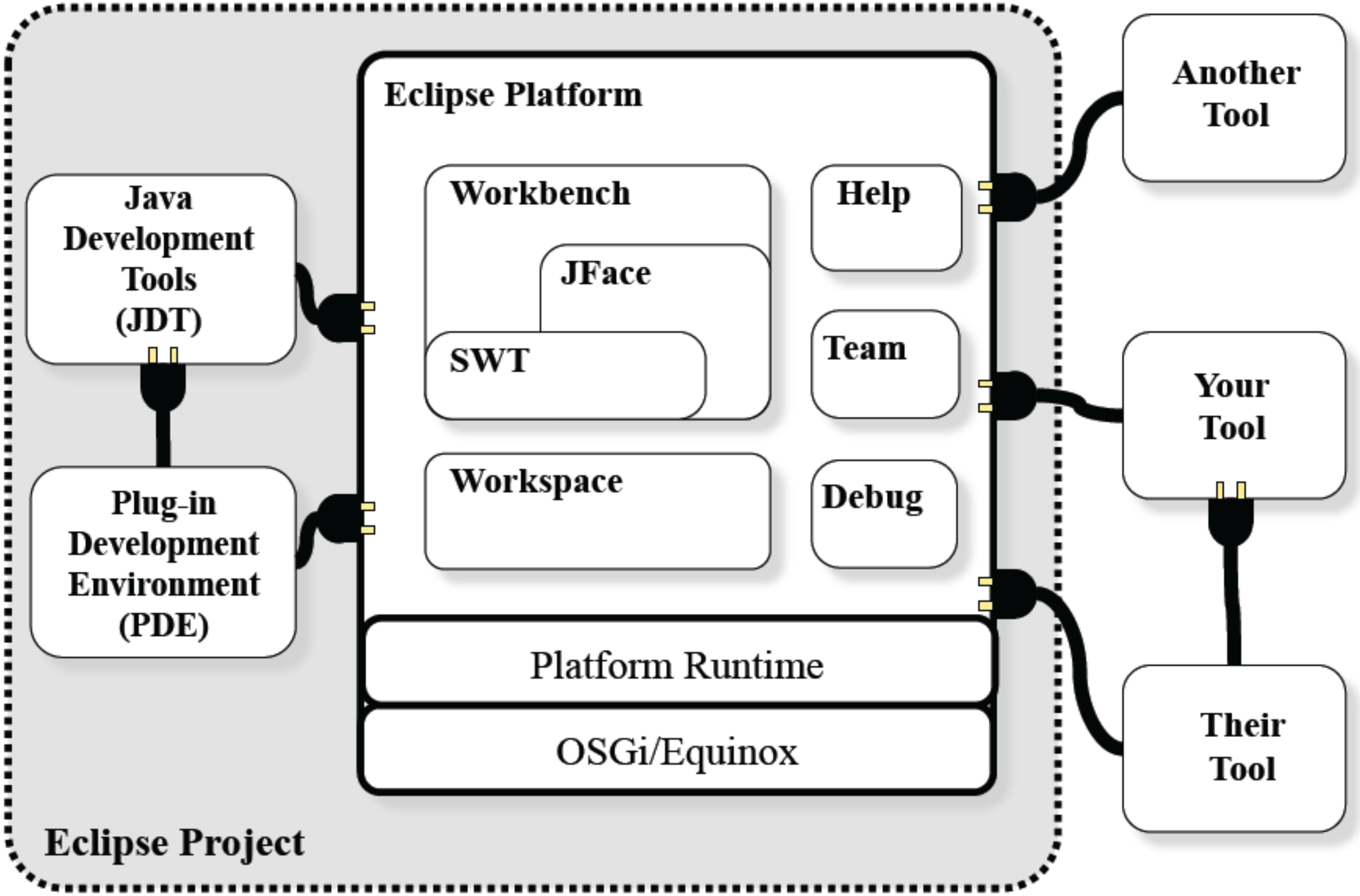
Everything is a plug-in

Plug-ins extend other plug-ins

- Creates dependencies

Does not have to be UI related

Architecture of Eclipse



Summary

Plug-ins need to be at a specific location and can be loaded at runtime

Plug-ins need description file of some sort

Generally, plug-in can modify displayed content after it is invoked

Plug-in gets control with callbacks

Each plug-in implements one or more well-defined interfaces

Next

Concepts

- Assignment 1: OSGI Lab (Friday + Monday)
- Assignment 2: Eclipse Plugin Lab (Tuesday + Thursday + Friday)

Things Due

- Assignment 1: Tuesday, 11:55 pm
- Assignment 2: Friday, 11:55 pm