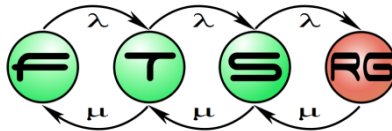


Bevezető

Eclipse alapú technológiák



Eclipse: Nyílt fejlesztőrendszer és platform

Modern fejlesztési folyamat

- Modell-alapú
- Több platform/több alkalmazás réteg
- Csoportmunka
- Több nyelvű
 - Több programozási nyelv
 - Több dokumentum-leíró nyelv
 - Több természetes nyelv (i18n)
- Nagy méretű projektek

Fejlesztési feladatok

- Modell szerkesztés
- Kód szerkesztés
- Fordítás, futtatás, debug
- Automatizált tesztek
- Adatbázis tervezés/implementálás
- Weboldalak készítése
- Dokumentáció!!!

Régi megközelítés

- Minden feladatra külön eszköz
 - Java nyelv: javac, javadoc, java, stb.
- Többnyire parancssori/szöveges eszközök
- Feladatváltás “fejben”
 - Nehézkes integráció
 - Nehéz mindent kézben tartani

- Nem elég hatékony

Trendek

- Kezdetek (1980-as évek első feléig)
 - Parancssori eszközök
 - Egy-egy részfeladat végrehajtása (pl. fordítás)
 - Sok ilyen eszköz
 - Fejlesztési folyamat részekre bomlik:
 - Szerkesztés, fordítás, debug külön-külön programokkal
 - Kezdetleges integráció
 - make

Trendek

■ 1980-as évek vége

○ Hagyományos toolok

- Már van IDE kezdemény (Turbo Pascal...)
- Csak egy gyártóhoz kapcsolódik
- Csak egy fejlesztési részfolyamathoz (pl. kódolás)

○ „4GL” (negyedik generációs eszközök)

- Speciális alkalmazások (pl. ügyviteli rendszerek) gyors fejlesztése (pl. Magic)
- Csak egy adott futtató környezetben működő alkalmazások

- 1990-es évek
 - Több különböző keretrendszer egy grafikus platformra
 - Win32
 - Borland – Visual Control Library (Delphi, C Builder)
 - MS – MS Foundation Classes (Visual Studio)
 - Linux
 - QT – QT Designer
 - KDE – KDevelop
 - Kylix (Borland)
 - Nehézkes az átjárás a toolok és platformok között

■ 1990-es évek

○ Java megjelenése

- Platform független (?)
- Objektum-orientált
- Egységes keretrendszer

○ Integrált fejlesztőeszközök nélkül reménytelen az ipari méretű alkalmazás fejlesztés

- Bonyolult platform (több száz osztály...)
- XML telepítési leírók (nem emberi editálásra optimalizálva...)

Trendek

- 2000 után
 - Sok komponensből álló keretrendszerek
 - Erős platform támogatás webes fejlesztéshez (.NET, J2EE)
 - Integrációs interfészek – bővíthetőség
 - Gyorsan változó platformok
 - Dinamikus változás a fejlesztőeszközökben is!
 - Jelentős kapacitást igényel a fejlesztőeszközök folyamatos frissítése
 - Felértékelődik a nyílt forráskód szerepe

Integrált fejlesztőrendszerek (IDE)

- Közös keret az eszközöknek
- Egy gyártó tool-jait fogja össze (pl. Borland Delphi, MS Visual Studio)
- Kényelmes, hatékony
- Külső eszközökkel nem vagy nehezen bővíthető ☹️
- Sok egyforma fejlesztési projekt esetén jó

Nyílt fejlesztőrendszerek

- Jól definiált keretrendszer
- Nyílt specifikáció
- Könnyen bővíthető
- Gyártó független
- (általában) platform-független
- (optimális esetben) programozási nyelv független
- Optimális a gyorsan változó igényekhez

- Integrált ÉS nyílt
 - Hasonló felület az eszközökhöz
 - Egyszerűen használhatunk többféle toolt egy projektben
 - Testre szabhatjuk – meglévő bővítményekkel
 - Testre szabhatjuk – saját bővítményekkel
 - 😊 A nyílt forráskódú közösség számára hatalmas lehetőségek

Néhány népszerű eszköz

- MS Visual Studio .NET
 - Integrált, nem nyílt...
 - Szinte csak MS technológiákat támogat
- Borland eszközök (Delphi, C Builder...)
 - Integrált, nem nyílt
 - Saját architektúra (a fejlesztéshez)
 - Több tanulást kíván
- Oracle Application Builder
 - Integrált, nem nyílt
 - Java alapú
 - Főleg Oracle támogatás

Néhány népszerű eszköz

■ Netbeans

- Nyílt
- Java-centrikus
- Bővíthető
- Ipari háttér (SUN)

■ Eclipse

- Nyílt
- Több nyelvet támogat (Java, C/C++, PHP,...)
- Erős ipari háttér (IBM, BEA, ...)

Miért indul ez a tárgy?

- A tendenciák a nyílt fejlesztőrendszerek előretörését mutatják (lásd később)
- Szükség van saját kiegészítések készítésére
 - Saját fejlesztések támogatására
 - Saját megoldások számára fejlesztőkörnyezet kialakítása
 - Üzleti céllal hiányzó funkciók implementálása
- Az országban is egyre nagyobb igény Eclipse programozókra!

Miért Eclipse?

- Erős ipari támogatottság
 - IBM, BEA, Oracle (részben)...
- Sok sikeres fejlesztés a világban (amiből lehet tanulni)
- Átgondolt architektúra
- Több programozási nyelv támogatása
- Több platform támogatása

Miért Eclipse?

- Szükség lehet Eclipse alapú alkalmazások fejlesztésére (Rich Client Platform)
 - Erős alapszolgáltatások grafikus alkalmazásokhoz
 - Egységes keretrendszer lehet több alkalmazás integrálására
 - Pl. az új Lotus Workspace

Eclipse: Nyílt fejlesztőrendszer

- Integrált fejlesztői környezet (IDE)
 - Több nyelv (Java, C/C++, PHP, ...)
 - Grafikus szerkesztők (pl. UML editor)
- Erős ipari háttér (IBM, Oracle, ...)
- Nyílt forráskód
- Átgondolt architektúra
 - Moduláris felépítés
 - Tervezési minták (Design patterns) alkalmazása
- Fejlesztés nyelve: Java

Eclipse: Fejlesztőkörnyezet

The screenshot displays the Eclipse IDE interface. The main editor shows the file `BankAccountTests.java` with the following code:

```
package org.eclipse.samples.banking;

import java.math.BigDecimal;

import org.junit.Test;
import static org.junit.Assert.*;

public class BankAccountTests {
    @Test
    public void testDeposit() throws Exception {
        BankAccount account = new BankAccount();
        account.deposit(new BigDecimal(1000));

        assertEquals("...", account.getBalance(), new BigDecimal(1000));
    }
}
```

A context menu is open over the `account.deposit` call, offering options: "Create method 'deposit(BigDecimal)' in type 'BankAccount'", "Add cast to 'account'", and "Rename in file (Ctrl+2 R direct access)". A yellow tooltip on the right shows the stub for the `BankAccount` class:

```
import java.math.BigDecimal;

public class BankAccount {

    public void deposit(BigDecimal bigDecimal) {
        // TODO Auto-generated method stub
    }
}
```

The bottom panel shows the "Problems" view with 3 errors:

Description	Resource	Path	Location
Syntax error, insert ";" to complete	BankAccountTests.java	BankingProject/src/org/eclipse/samples/banking/BankAccountTests.java	line 15
The method deposit(BigDecimal) is undefined for the type BankAccount	BankAccountTests.java	BankingProject/src/org/eclipse/samples/banking/BankAccountTests.java	line 13
The method getBalance() is undefined for the type BankAccount	BankAccountTests.java	BankingProject/src/org/eclipse/samples/banking/BankAccountTests.java	line 15

The status bar at the bottom indicates: "The method deposit(BigDecimal) is undefined for the type BankAccount | Writable | Smart Insert | 13 : 18".

Eclipse: Alkalmazásplatform

■ Rich Client Platform

- Eclipse technológiákon alapuló asztali szoftver platform
- IDE bővítmények újrafelhasználhatóak
- Egységes keretrendszer lehet alkalmazásintegrációhoz (pl. Lotus Symphony irodai szoftver)

■ Rich Ajax Platform

- Eclipse technológiákon alapuló webes platform
- Cél: Single Sourcing
 - RCP alkalmazások minél teljesebb újrahasznosítása
- Alternatíva lehet J2EE-vel szemben

RCP alkalmazás - XMIND

The screenshot displays the XMIND 2007 application window. The main workspace shows a mind map with a central node 'Sitemap' (blue oval) and several branches:

- Solutions** (blue circle): Includes sub-nodes for 'Individual' (Presentation, Decision Maker, Writing Helper, Time Management), 'Education' (Innovation, Teaching, Key Notes), and 'Teamwork' (Brainstorming).
- Products** (orange oval): Includes sub-nodes for 'Home' (What's New), 'Overview', 'Screenshots', 'Tech Notes', 'Downloads', and 'Purchase'.
- Features** (green oval): Includes a list of 11 features:
 1. Structure for human brains
 2. Rich expression
 3. Boundary and Relationship
 4. Markers and Legend
 5. Strong auto-numbering
 6. Customized Templates
 7. Open Eclipse Plug-in Platform
 8. Filter and delamination
 9. Powerful workbook and asso
 10. Seamless integration with of
 11. Import other mindmaps to sa
- Support** (blue globe): Includes sub-nodes for 'FAQ', 'Install', and 'Forum'.
- About** (yellow notepad): Includes sub-nodes for 'Pictures', 'Map', and 'Privacy'.
- Next Version** (white oval): Includes sub-nodes for 'Solutions', 'Tech Notes', 'Downloads', and 'Cases'.

The 'Analytics' node (pink cloud) is connected to 'Next Version' via a dotted line. It includes sub-nodes for 'Pageviews', 'Visits', and 'PV', with a note 'By Google' above it.

The right sidebar contains several panels: 'Outline' (showing a tree view of the map), 'Notes', 'Markers' (with a filter), and 'XMIND Markers' (with various icons like Priority, Smiley, Task Complete, and Flags).

The bottom panel shows a 'Properties' section and a 'Templates' section with six options: Default Template, XMIND Classic, XMIND Simple, XMIND Business, XMIND Academese, and XMIND Comic.

The status bar at the bottom indicates '1 topic. (# Sitemap *) selected.'

RCP alkalmazás - Kalypso

The screenshot displays the Kalypso software interface. The main window shows a map of a catchment area with subcatchments highlighted in red. The interface includes a Navigator, Style Editor, and a data table.

Style Editor: Subcatchments

Regel: + + - ←

Subcatchments | Subcatchments-Numb

Titel: Subcatchments

MinDenom: 0.0

MaxDenom: 77277.838217127

Symbolizer: Polygo Ort

Legende:

Polygon | Line | Point | Text

Füllung

Fill-Farbe:

Fill-Opacity: 0.2

Outline:

+ Legende

+ Knoten

+ VGewaesser

+ KMGewaesser - aktiv

+ RHBGewaesser

Teilgebiete

+ Subcatchments

Hydrotoper

+ hydrotop

Table: *Tabelle_Teilgebiete.gtt

TG-Nummer (in...	Versiegelu...	Anstie...	TG-Flaeche (fla...	Faktor ...	Anfangsinhalt...
100	0.188		1123080		
101	0.0080				
102					
				2.0	
				1.5	

<http://www.kalypso-simulation-platform.org>

RAP alkalmazás – Yoxos OnDemand

The screenshot shows the Yoxos OnDemand web application interface. The browser address bar displays `http://ondemand.yoxos.com` and the page title is "Yoxos OnDemand – Get your personalized Eclipse". A warning message states: "This web site does not supply identity information." The main header features the "yoxos on demand" logo and the text "free eclipse download service".

The interface is divided into several sections:

- Components:** A sidebar on the left lists various Eclipse components for selection, including "Managed Templates", "C and C++ Development", "Charting and Reporting", "Communications", "Database Development", "Desktop", "Eclipse Development", "Graphical Editors and Frameworks", "Java Development", "Mobile Java", "Models and Model Development", "Mylyn", "Other Tools", "Programming Languages", "Quality Assurance", "Remote Access and Device Development", "Runtime", "SOA Development", "Science", "Source Code Management", "Sources", "Testing and Performance", "UI Development", "Web and Java EE Development", and "Yoxos Tools".
- Plan:** The central area shows a dropdown menu set to "MacOS X Cocoa". Below it, a message states: "No components are selected for installation. Use the Add button to insert contents into the Plan." There are "Add Shared Template", "Save As Template", and "Start Download" buttons. The estimated download size is "0 kB".
- Information:** A section at the bottom of the main area with the text: "Select elements from other views to display their information."
- Get Certified Components:** A promotional banner for "Yoxos SecunSource" with a "Feedback" button and the text: "At a special introductory price until December 1st."
- Cheat Sheets:** A section on the right with the title "Yoxos OnDemand: Free Eclipse Download Service" and an "Introduction" section. The introduction text reads: "Discover the Yoxos OnDemand Services: Create a custom download by selecting your favorite plugins or create a profile consisting of plugins AND team project sets, preferences, mylyn queries and more. We combine everything into a single download that you simply extract to create your personal Eclipse installation." Below this is a "Click to Begin" button and a list of steps: "Selecting your Operating System", "Getting", and "Automated dependency".

Adminisztráció

Adminisztráció

■ Órák

- Szerda: 12:15-14:00 előadás,
- Péntek: 12:15-14:00 labor, BME IE224

■ Követelmények

- Házi feladat
- Vizsga

Házi feladat

- Eclipse alapú fejlesztés végzése
 - Plug-in
 - (OSGi alkalmazás)
 - RCP alkalmazás
 - RAP alkalmazás
- Témalista a weben
- Értékelés
 - Függ a választott feladat nehézségétől
 - Felhasznált Eclipse alapú technológiától

Házi feladat

- Csapatmunka
 - Indokolt esetben lehetséges
 - 2-3 fő, behatárolt részfeladatok!
- Ütemezés
 - 4. hét végére: részletes specifikáció
 - Utolsó oktatási hét: beadás

Házi feladat

- Saját ötlet is lehet
 - Előzetes egyeztetés szükséges
 - Specifikációt részletesebben kell elkészíteni
 - Eredeti ötlet értékelése pozitív
 - Gyakorlatban használható: nagyon pozitív 😊

- Eclipse honlap
 - www.eclipse.org
 - Wiki: wiki.eclipse.org
 - További linkek Eclipse oldalakhoz
- Tárgy segédanyagai
 - Előadás fóliák
 - Egyéb segédanyagok
 - <http://eclipse.inf.mit.bme.hu>
 - Fejlesztés alatt

Eclipse – Az első lépések

Letöltés

- www.eclipse.org/downloads
- Ingyenesen letölthető keretrendszer
 - Legtöbb platformra elérhető
- Legfrissebb verzió
 - Eclipse 3.5.2

Letölthető csomagok

- Többféle változat letölthető
 - Eclipse IDE for Java EE Developers
 - Eclipse IDE for Java Developers
 - Eclipse for PHP Developers
 - Eclipse IDE for C/C++ Developers
 - **Eclipse for RCP/Plug-in Developers**
 - Eclipse Modeling Tools
 - ...
- Ha nem megfelelőt választottunk, utólag “javítható” bővítmények telepítésével

Java környezet

- Java SDK szükséges
 - <http://java.sun.com/javase/downloads/index.jsp>
 - Ajánlott verzió: Java 6
- Keretrendszer
 - Tömörített állomány
 - Bárhova kicsomagolható és futtatható

Workspace

- Első indításkor workspace helye:
 - Egy könyvtár a merevlemezen
 - A felhasználói adatokat tartalmazza
 - Projektek
 - Mappák
 - Fájlok
 - Beállítások

Első házi feladat

- Eclipse 3.5 telepítése
 - Segíteni fog a félév során 😊

Eclipse for RCP/Plug-in Developers

- Telepített komponensek
 - Java Developer Tools (JDT)
 - Java fejlesztői eszközök
 - Plug-in Developer Tools (PDT)
 - Támogatás Eclipse bővítmények készítéséhez
 - Rich Client Platform
 - Alkalmazásfejlesztési támogatás
 - Mylyn, CVS
 - Csoportmunkatámogatás
 - Help
 - Súgó

Grafikus felület

Workbench

- Általános felületdefiníció
 - Felhasználó könnyen testre szabhatja
- Elemek
 - Ablak
 - Lap
 - Perspektíva
 - Nézetek
 - Szerkesztők

Workbench felépítése

Workbench window

The screenshot displays the Eclipse IDE Workbench window. The title bar reads "Java - hu.optxware.eclipsecourse.rcpdemo.gui/src/hu/optxware/eclipsecourse/rcpdemo/gui/ServiceInterface.java - Eclipse SDK - /Users/stampie/Document...". The interface includes a Package Explorer on the left, a central Editor showing the ServiceInterface.java file, an Outline view on the right, and a Problems view at the bottom. A red speech bubble points to the Package Explorer with the text "Workbench window".

Package Explorer:

- hu.optxware.eclipsecourse
- hu.optxware.eclipsecourse.rcpdemo.gui
- JRE System Library [Java]
- Plug-in Dependencies
- src
 - hu.optxware.eclipsecourse.rcpdemo.gui
 - Activator.java
 - Application.java
 - ApplicationActionBarAdvisor.java
 - ApplicationWorkbenchAdvisor.java
 - ApplicationWorkbenchWindowAdvisor.java
 - BookListView.java
 - BookListViewContentProvider.java
 - Perspective.java
 - ServiceInterface.java
 - ServiceManager.java
 - hu.optxware.eclipsecourse.rcpdemo.gui.comr
 - ExitHandler.java
 - icons
 - META-INF
 - OSGI-INF
 - build.properties
 - plugin.xml
 - splash.bmp
- hu.optxware.eclipsecourse.rcpdemo.model.simple

Editor: ServiceInterface.java

```
ServiceInterface {  
    serviceCreated();  
    serviceRemoved();  
}
```

Outline:

- hu.optxware.eclipsecourse
 - ServiceInterface
 - serviceCreated() : void
 - serviceRemoved() : void

Problems: 0 items

Description	Resource	Path	Location	Type
-------------	----------	------	----------	------

Bottom Bar: Writable | Smart Insert | 1 : 1

Workbench felépítése

The screenshot displays the Eclipse IDE interface. The main toolbar is highlighted with a blue box. A callout bubble with a dark red background and white text points to this toolbar, containing the text "Main toolbar". The IDE shows a project named "hu.optxware.eclipsecourse.rcpdemo.bl.simple" with a source folder "src" containing several Java files, including "ServiceInterface.java". The "Outline" view on the right shows the "ServiceInterface" interface with methods "serviceCreated() : void" and "serviceRemoved() : void". The "Problems" view at the bottom shows 0 items. The status bar at the bottom indicates "Writable", "Smart Insert", and "1 : 1".

Main toolbar

```
package hu.optxware.eclipsecourse.rcpdemo.gui;

public interface ServiceInterface {
    void serviceCreated();
    void serviceRemoved();
}
```

Description	Resource	Path	Location	Type

Workbench felépítése

The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows a project structure with the following packages and files:

- hu.optxware.eclipsecourse.rcpdemo.bl.simple
- hu.optxware.eclipsecourse.rcpdemo.gui
 - JRE System Library [Java]
 - Plug-in Dependencies
 - src
 - hu.optxware.eclipsecourse.rcpdemo.gui
 - Activator.java
 - Application.java
 - ApplicationActionBarAdvisor.java
 - ApplicationWorkbenchAdvisor.java
 - ApplicationWorkbenchWindowAdvisor.java
 - BookListView.java
 - BookListViewContentProvider.java
 - Perspective.java
 - ServiceInterface.java
 - ServiceManager.java
 - hu.optxware.eclipsecourse.rcpdemo.gui.comr
 - ExitHandler.java
 - icons
 - META-INF
 - OSGI-INF
 - build.properties
 - plugin.xml
 - splash.bmp
- hu.optxware.eclipsecourse.rcpdemo.model.simple

The main editor window shows the code for `ServiceInterface.java`:

```
package hu.optxware.eclipsecourse.rcpdemo.gui;  
  
public interface ServiceInterface {  
    void serviceCreated();  
    void serviceRemoved();  
}
```

The Outline view on the right shows the project structure:

- hu.optxware.eclipsecourse
 - ServiceInterface
 - serviceCreated() : void
 - serviceRemoved() : void

The Problems view at the bottom shows 0 items.

Description	Resource	Path	Location	Type

At the bottom of the IDE, the status bar shows: Writable | Smart Insert | 1 : 1

Workbench Site

- Egy ablak egy lapot tartalmaz
- Lap
 - Fő tartalom
 - Szerkesztők
 - Nézetek
 - Elrendezés perspektíva alapján

Workbench felépítése

The screenshot shows the Eclipse IDE interface. On the left is the Package Explorer showing the project structure. The main editor displays the `ServiceInterface.java` file with the following code:

```
package hu.optxware.eclipsecourse.rcpdemo.gui;  
  
public interface ServiceInterface {  
    public void serviceCreated();  
    public void serviceRemoved();  
}
```

A red callout bubble points to the Java icon in the top toolbar, with the text "Perspektíva választó" (Perspective selector).

At the bottom, the Problems view is visible, showing 0 items. The status bar at the bottom indicates "Writable", "Smart Insert", and "1 : 1".

Perspektívák

- Nézetek funkcionális csoportja
- Elrendezést is rögzít
- Általában feladat-centrikus összeállítás
- Példák
 - Java Development
 - Plug-in Development
 - Debug
 - Team Synchronizing

Workbench felépítése

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project structure with several Java files, including `ServiceInterface.java`. The main editor window shows the content of `ServiceInterface.java`, which defines a public interface with two methods: `serviceCreated()` and `serviceRemoved()`. A blue rectangular box highlights the editor area. Two callout boxes point to this area: one on the left labeled "Szerkesztők (0..N)" and one on the right labeled "Megnyitott fájlok". The bottom of the IDE shows the Problems, Javadoc, and Declaration tabs, with the Problems tab currently displaying 0 items.

```
package hu.optxware.eclipsecourse.rcpdemo.gui;

public interface ServiceInterface {

    public void serviceCreated();

    public void serviceRemoved();

}
```

Description	Resource	Path	Location	Type

Szerkesztők

- Szerkesztési feladatokra
 - Tipikus: fájl szerkesztő
 - Egy vagy több fájl
 - DE: bármi fölé lehet szerkesztő
 - Pl. adatbázistartalom
 - Pl. verzió összehasonlítás (compare editor)
- Lehet
 - Szöveges (pl. Java editor)
 - Űrlap-alapú
 - Grafikus (pl. UML editor)
 - Összetett (több lap, laponként különböző)

Workbench felépítése

The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows the project structure, with the `ServiceInterface.java` file selected under the `src` directory. The main editor window shows the code for the `ServiceInterface` class, which is a public interface with two methods: `serviceCreated()` and `serviceRemoved()`. The Outline view on the right shows the class hierarchy, including the `ServiceInterface` class and its methods. The Problems view at the bottom shows 0 items.

```
package hu.optxware.eclipsecourse.rcpdemo.gui;

public interface ServiceInterface {

    public void serviceCreated();

    public void serviceRemoved();

}
```

Nézetek (0..N)
csoportokban

Workbench felépítése

The screenshot displays the Eclipse IDE interface. On the left, the Package Explorer shows a project hierarchy for 'hu.optxware.eclipsecourse.rcpdemo.gui'. The file 'ServiceInterface.java' is highlighted. The main editor window shows the source code of 'ServiceInterface.java', which defines a public interface with two methods: 'serviceCreated()' and 'serviceRemoved()'. The Outline view on the right shows the project structure, including the 'ServiceInterface' interface and its methods. A callout box with the text 'Projekt hierarchia' points to the 'ServiceInterface.java' file in the Package Explorer.

```
package hu.optxware.eclipsecourse.rcpdemo.gui;

public interface ServiceInterface {

    public void serviceCreated();

    public void serviceRemoved();

}
```

Projekt hierarchia

Description	Resource	Path	Location	Type
items				

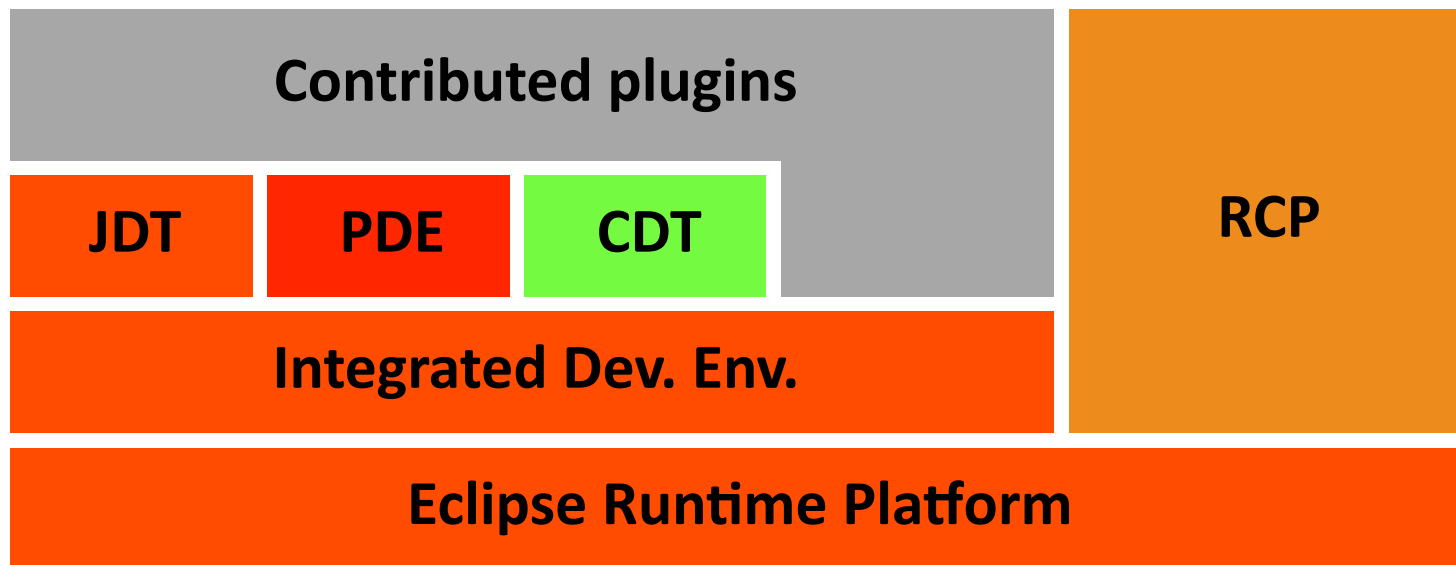
Nézetek

■ Kiegészítő funkciók

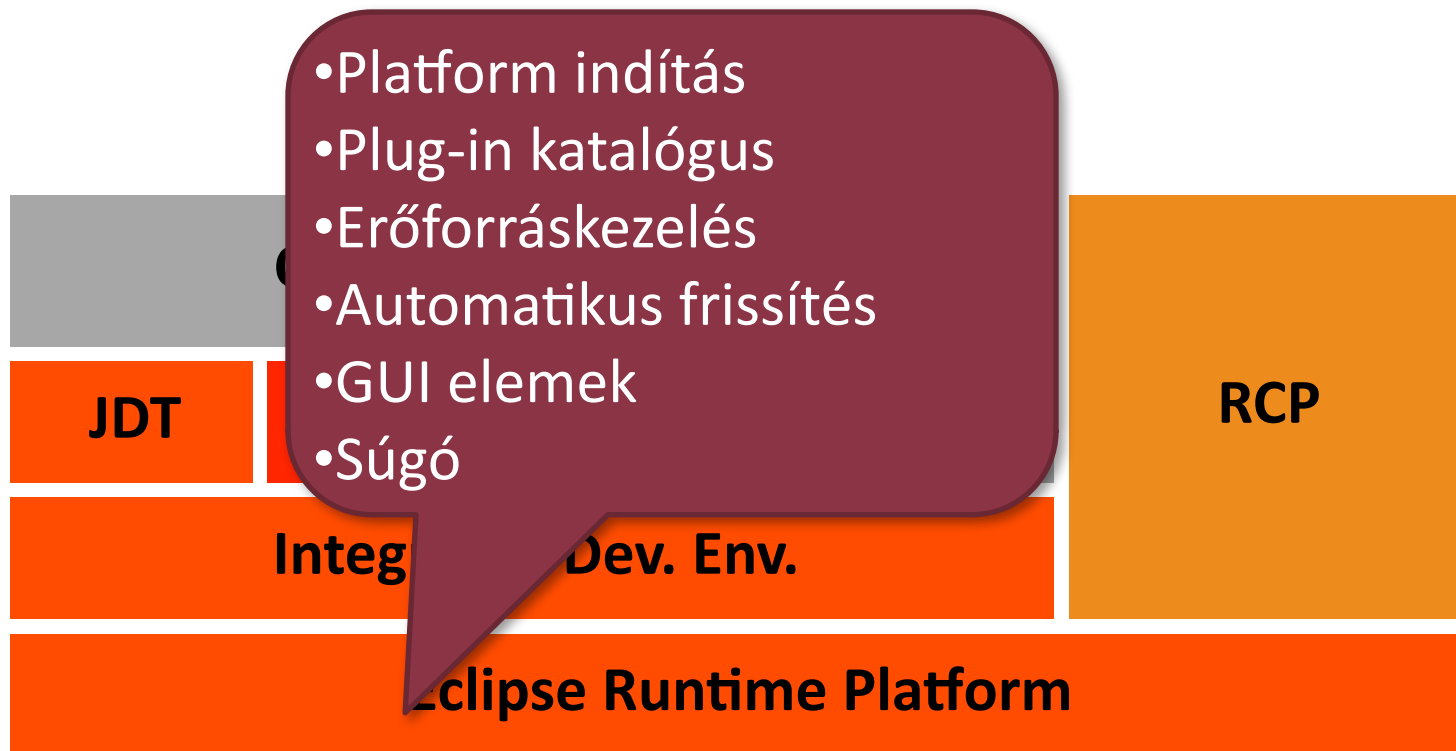
- Statikus információmegjelenítés
- Lehet globális
 - Erőforrásfa (Project Explorer)
 - Üzenetsor (Error Log)
- Vagy függhet a kijelölt elemektől
 - Tulajdonságok nézet (Properties – kijelölt elem tulajdonságai)
 - Áttekintő (Outline – aktuális editor tartalmának szerkezete)

Eclipse architektúra

Architektúra



Architektúra



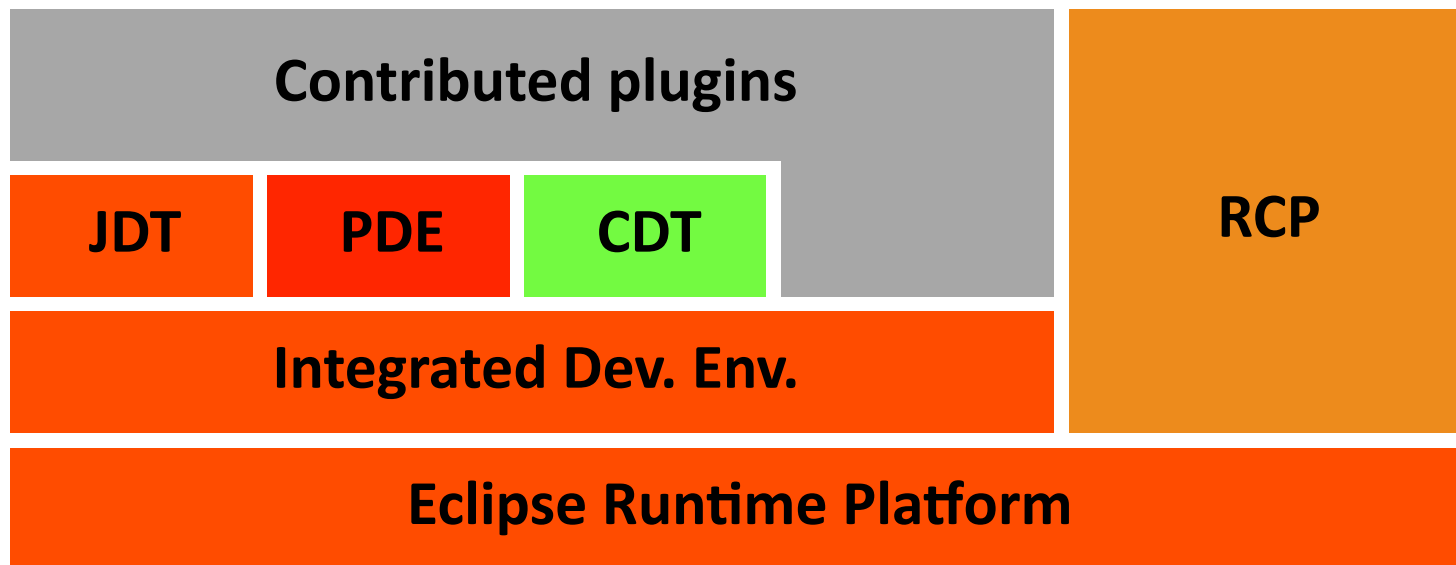
Runtime platform

- Plug-in katalógus
 - Plug-inek adatainak gyűjteménye
 - Függőségek kezelése
 - Plug-inek betöltése
 - Támogatja a lazy loading-ot
- Platformfüggetlen erőforráskezelés
 - Alapelem: workspace
 - Tartalom elérés workspace-relatív módon
 - Módosítások követése (refresh)
 - Karakterkódolás és fájlkezelés
 - Külső fájlok, erőforrások linkelhetőek a workspace alá

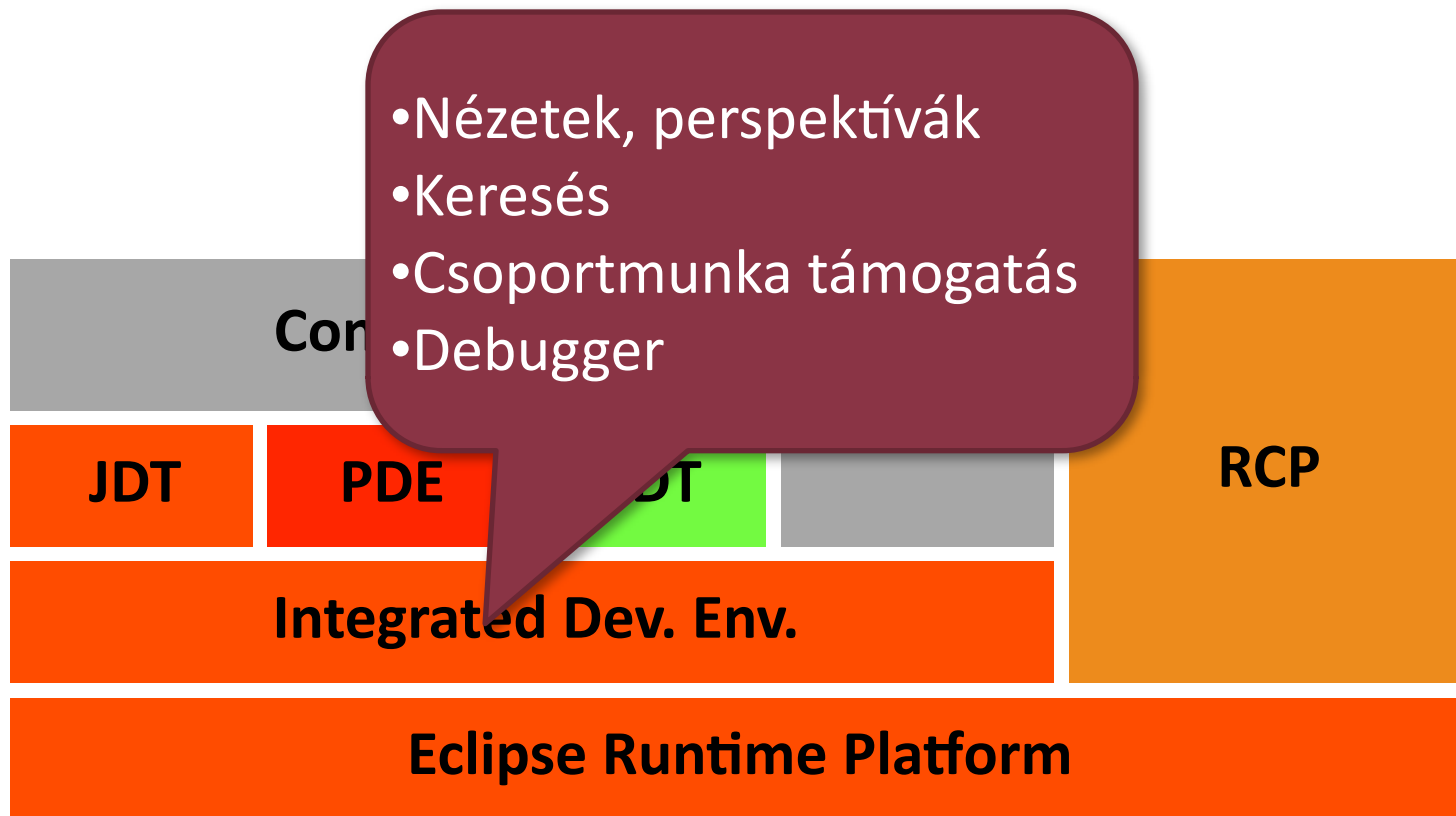
Runtime platform

- Automatikus frissítés
 - Plug-inek mellé definiálhatók „update site”-ok
 - Keretrendszer figyeli ezeket
 - Frissítések letölthetőek
 - Frissítést a kliensoldal kezdeményezi
- GUI elemek
 - SWT: egyszerű, natív GUI komponensek
 - JFace: SWT-re épülő, bonyolultabb elemek készítésére
 - Később részletesebben
- Súgó
 - HTML-alapú rendszer
 - XML leírókkal frissíthető
 - Szintén plug-in alapú

Architektúra



Architektúra



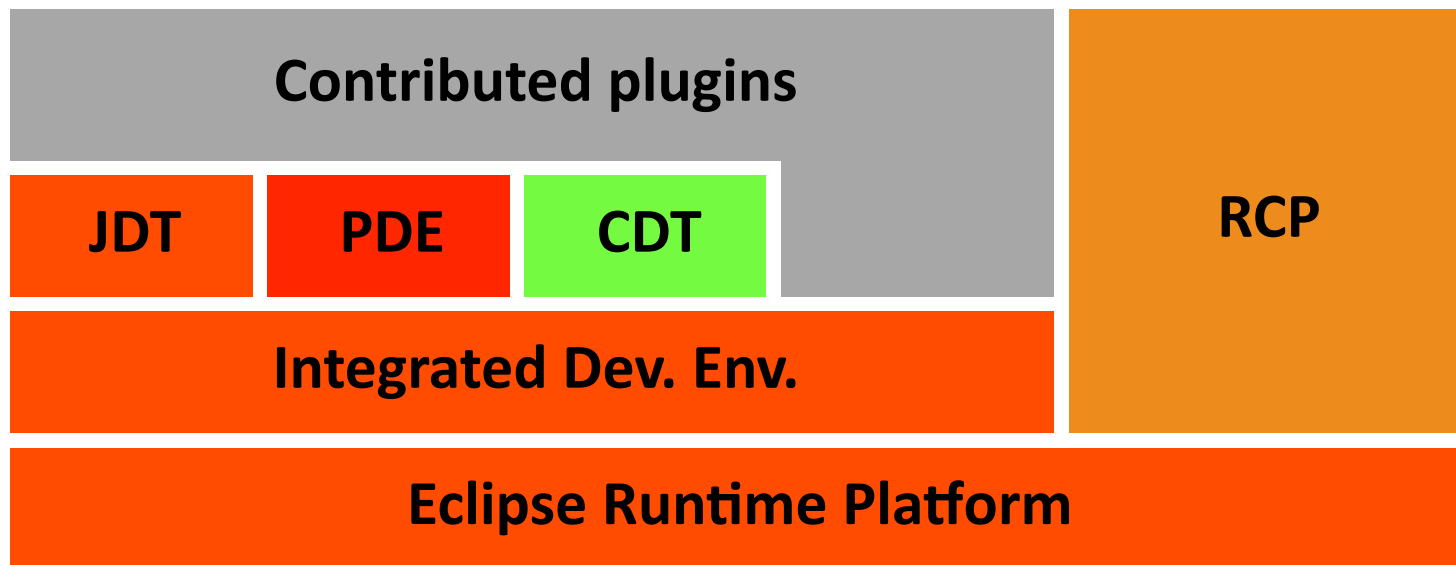
Integrált fejlesztőkörnyezet

- Funkciók absztrakt megvalósítása
 - Absztrakt komponensek
 - Nyelvfüggetlenség
 - Eszközfüggetlenség
 - Egységesen kezelhetőek a csatolt eszközök
 - Csak műveletdefiníciók
 - Pl. debuggernél: program léptetése
 - Pl. szerkesztőnél: fordítás

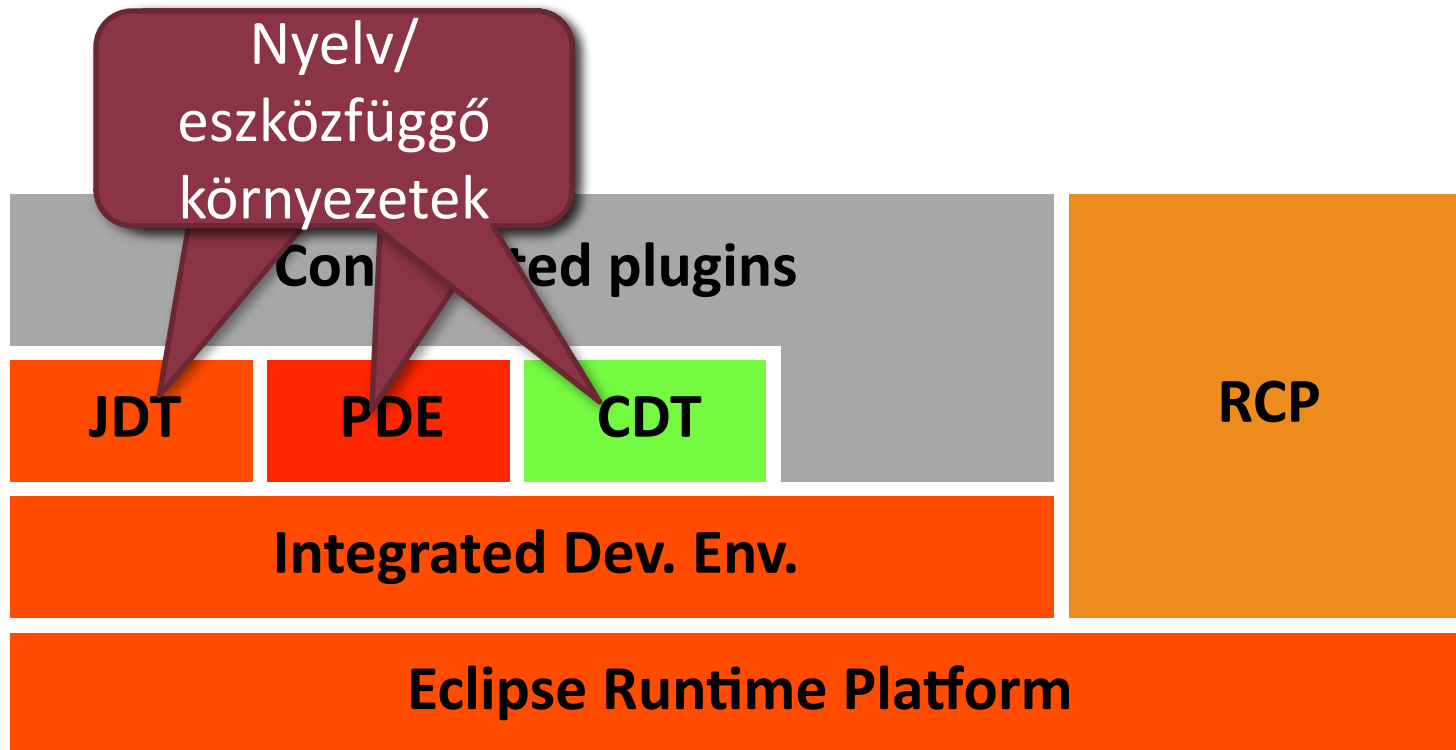
Integrált fejlesztőkörnyezet

- Keresés
 - Keresés futtatása
 - Eredmény megjelenítése
- Csoportmunka támogatás
 - Általános felület verziókezelő rendszerek illesztésére
 - Verzió történet (History)
 - Verziók összehasonlítása (Compare editor)
- Debugger
 - Általános erőforrásokra definiálva
 - Memória, regiszterek, stack frame, stb.
 - Töréspont, kifejezések figyelése, stb.
 - Parancsok: indítás, léptetés...

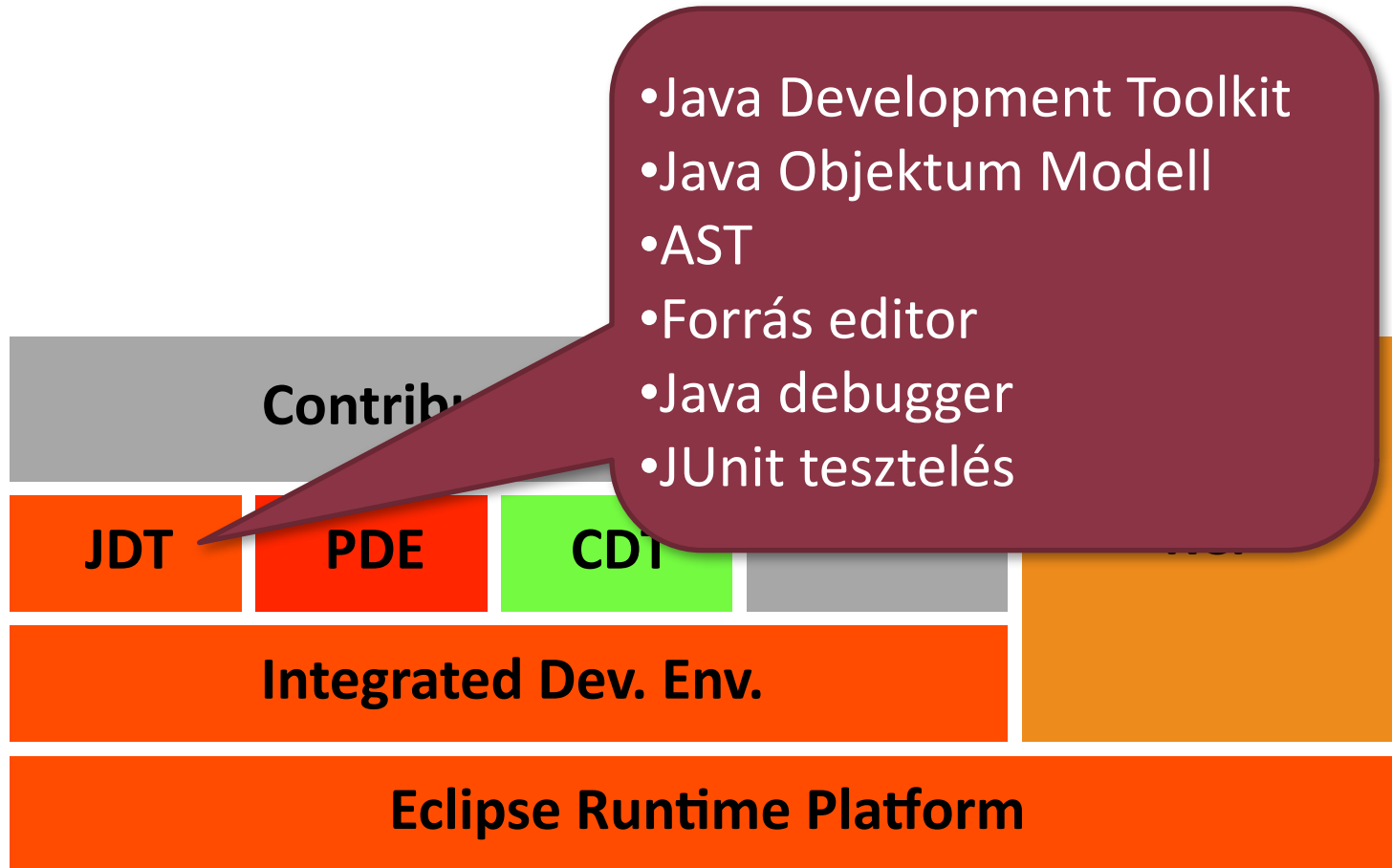
Architektúra



Architektúra



Architektúra



Java Development Toolkit (JDT)

- Java fejlesztői környezet
 - Teljes Java 6.0 támogatás
- Funkciók
 - Java szerkesztő
 - Automatikus ellenőrzéssel
 - Saját, inkrementális fordító
 - Java debugger
 - Általános komponens alapján
 - Unit teszt támogatás (JUnit)
 - Kiterjeszthető

Objektum modell és szintaxisfa

■ Felhasználás

○ Forráskód manipuláció

- Refactoring
- Automatikus kód transzformáció
- Kódgenerálás

○ Forráskód analízis

- Metrikák (pl. külső Eclipse Metrics plug-in)
- Statikus analízis (pl. külső FindBugs plug-in)
- Kód formázási szabályok (pl. külső CheckStyle plug-in)

Java Object Model

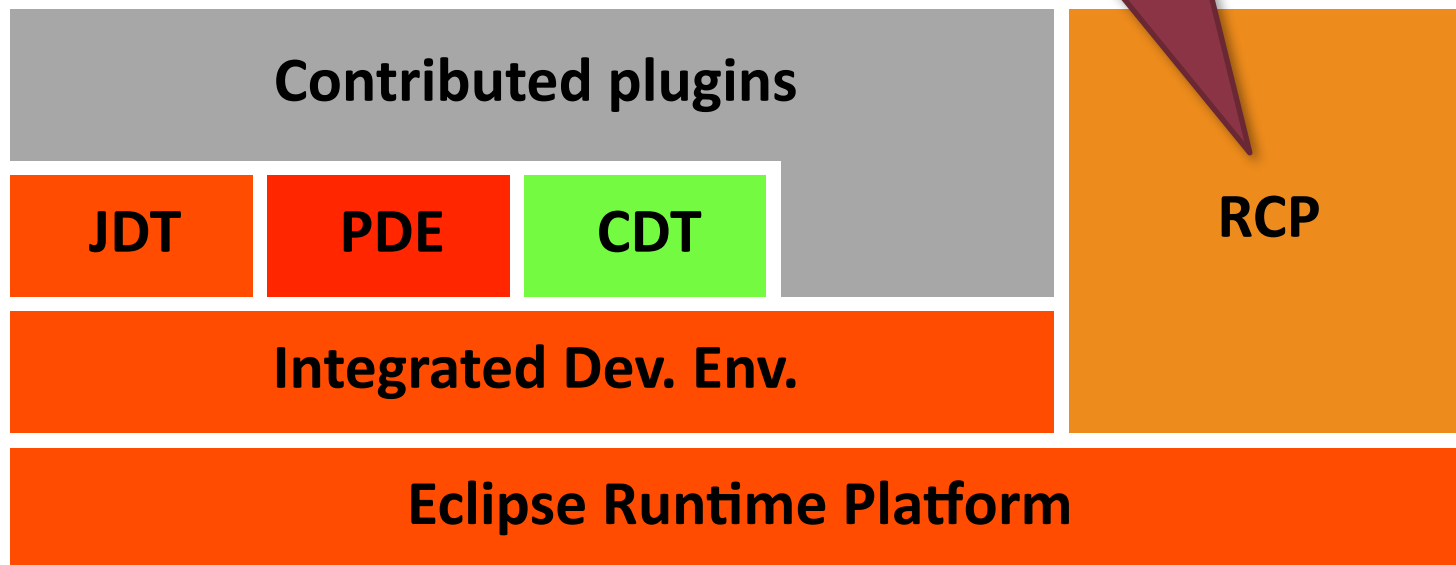
- A model
 - Egy Java program elemeit tárolja
 - Bináris (.class) és forrásfájlokat is kezel
 - Metódus tartalma nincs modellezve
 - Egyszerű szöveggént lekérhető
 - Hierarchikus felépítésű
 - Lekérdezés és manipuláció is lehetséges

Abstract Syntax Tree

- Metódusok belseje: Abstract Syntax Tree
 - Utasítás-szintig lebontja a forráskódot
 - Cél: elemi forráskód analízis és manipuláció
 - Miért nem ilyen a Java Object Model?
 - Erőforrás-takarékosság
 - Sebesség
 - Egy tetszőleges Java elem kibontható (osztály, metódus...)
 - Tartalmaz referenciákat
 - Típusra, metódusra, mezőre
 - Lusta kifejtés
 - Így gyorsabb

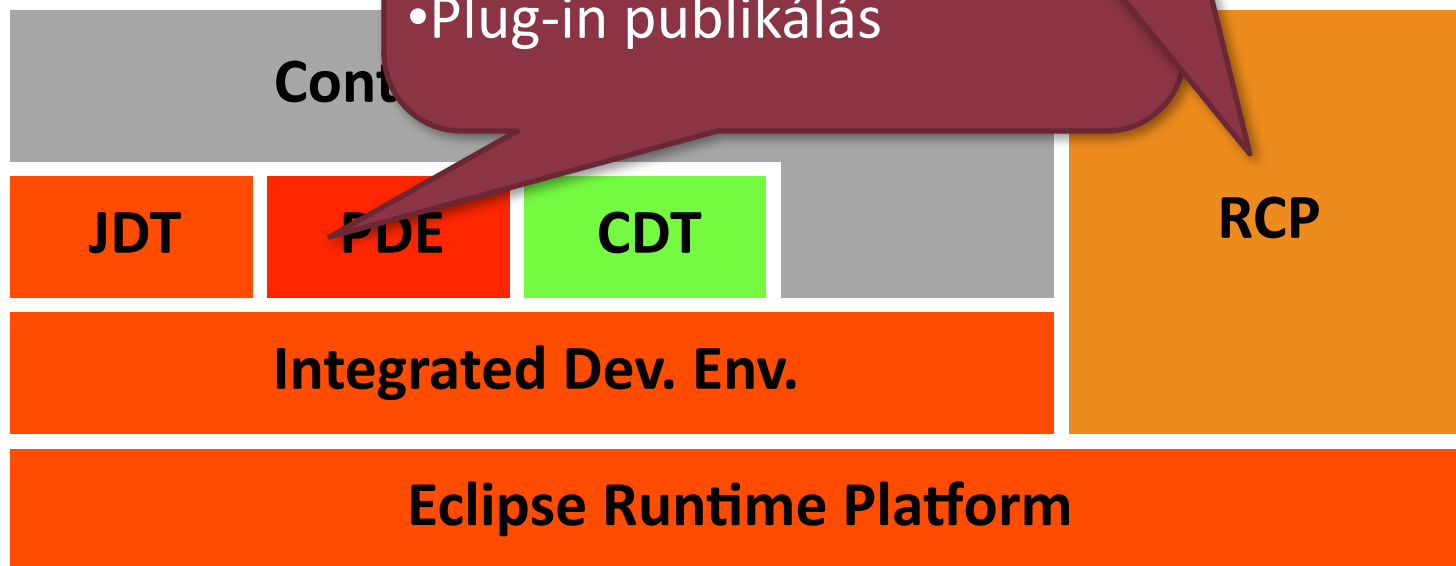
Architektúra

- Alkalmazások
- Runtime Platform felett

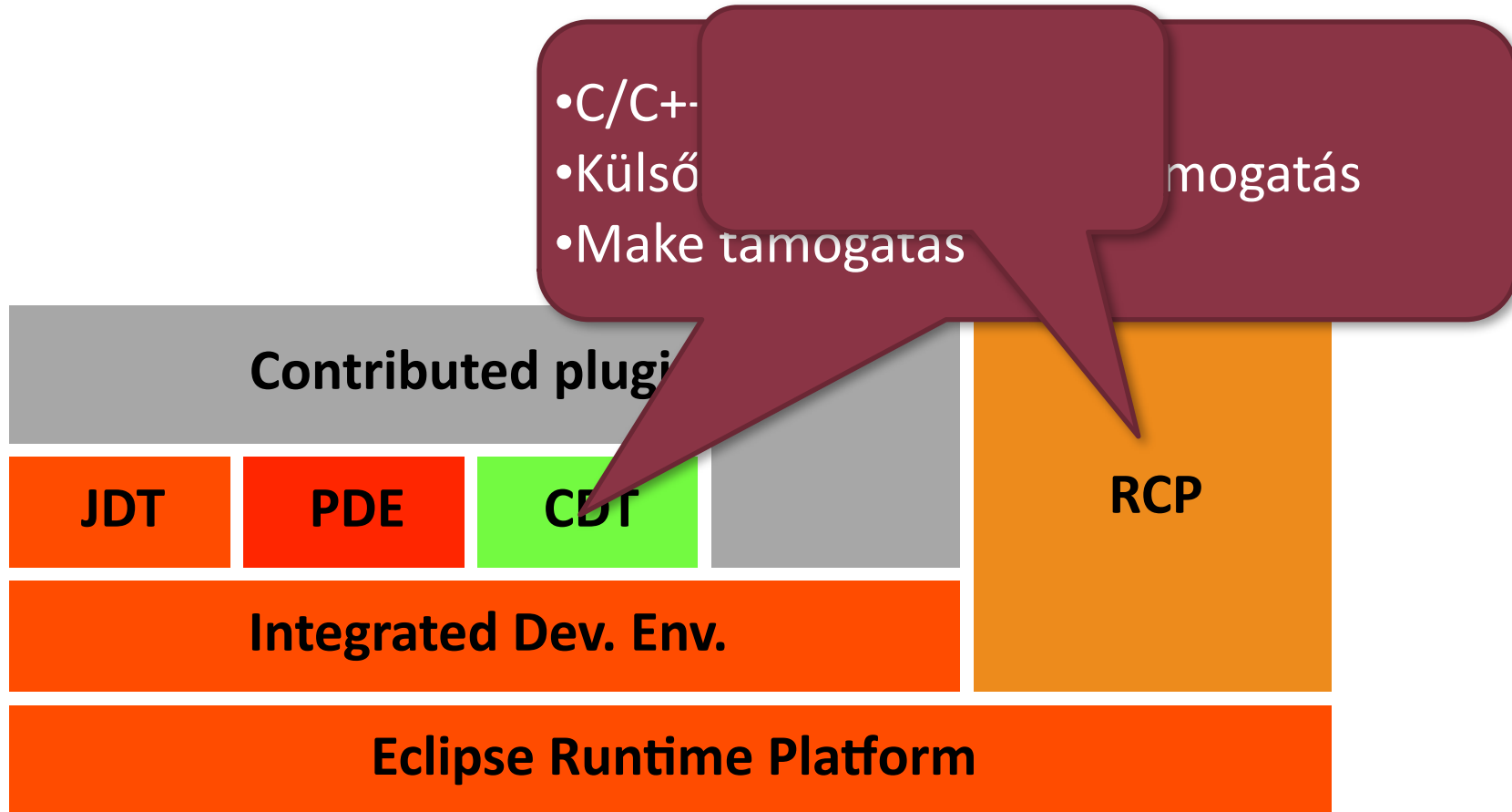


Architektúra

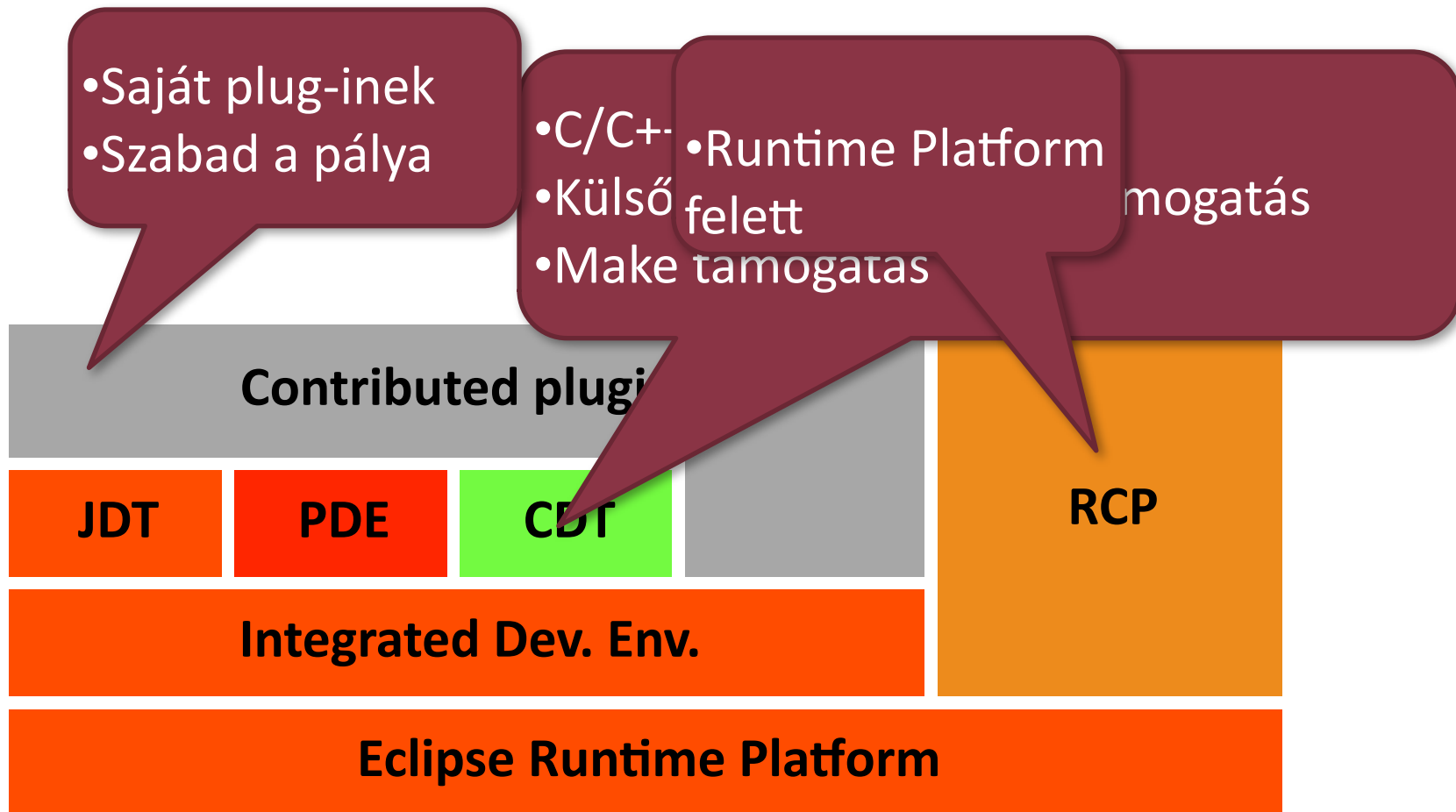
- Saját plug-in
- Futtatás támogatás
- Plug-in definíció készítés
- Plug-in publikálás



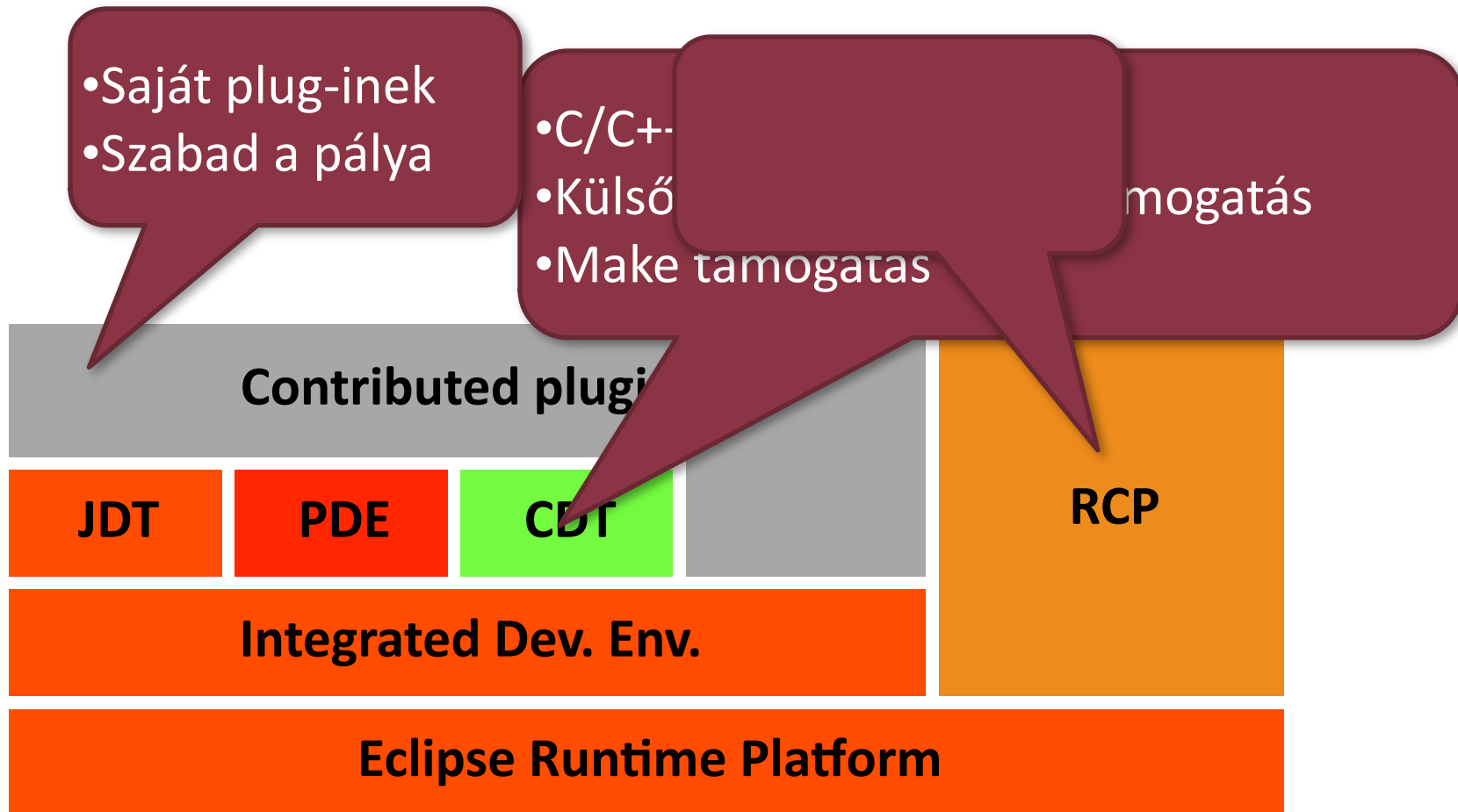
Architektúra



Architektúra



Architektúra



Összefoglalás

- Eclipse fejlesztés
 - Plug-inek készítése
 - Java nyelven
 - Cél:
 - eszközök összehangolása