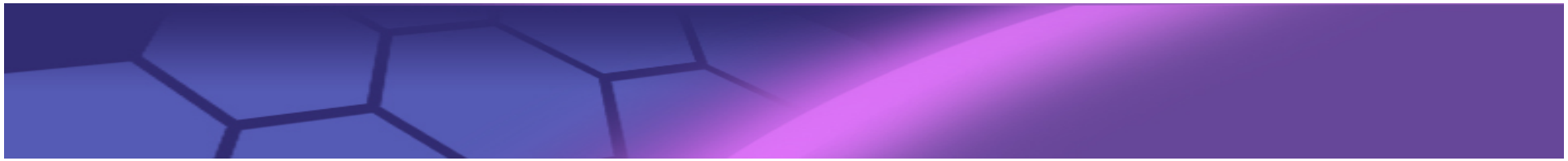
Business Intelligence & Reporting Tools (BIRT) Mastering BIRT

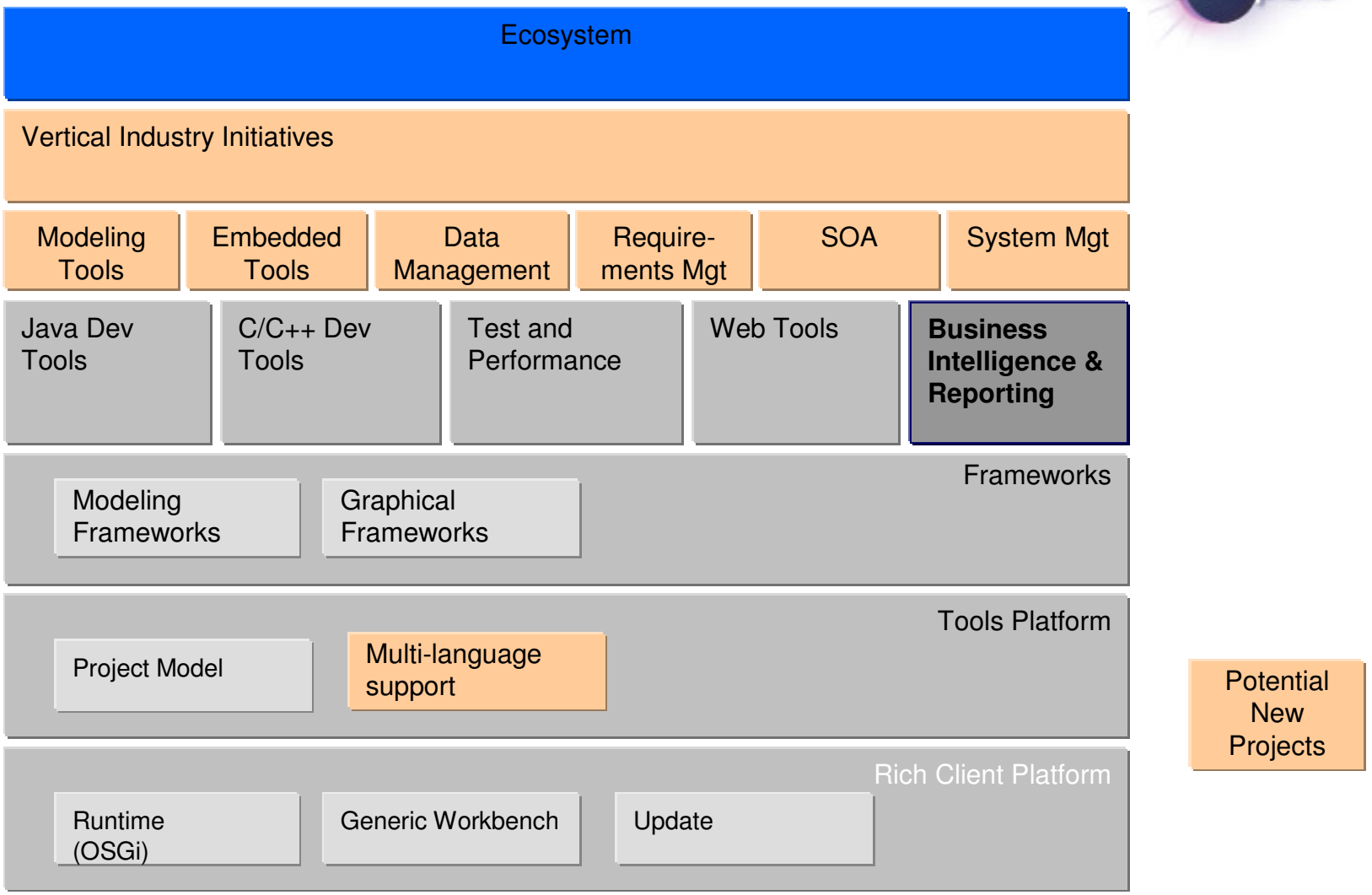
Scott Rosenbaum

BIRT Project Management Committee

Innovent Solutions, Inc.



BIRT in the Big Picture



BIRT Top-Level Project Scope

Operational Reporting	Ad hoc Query & Reporting	Analytics/OLAP/Data Mining

In Reality, this is a Continuum:

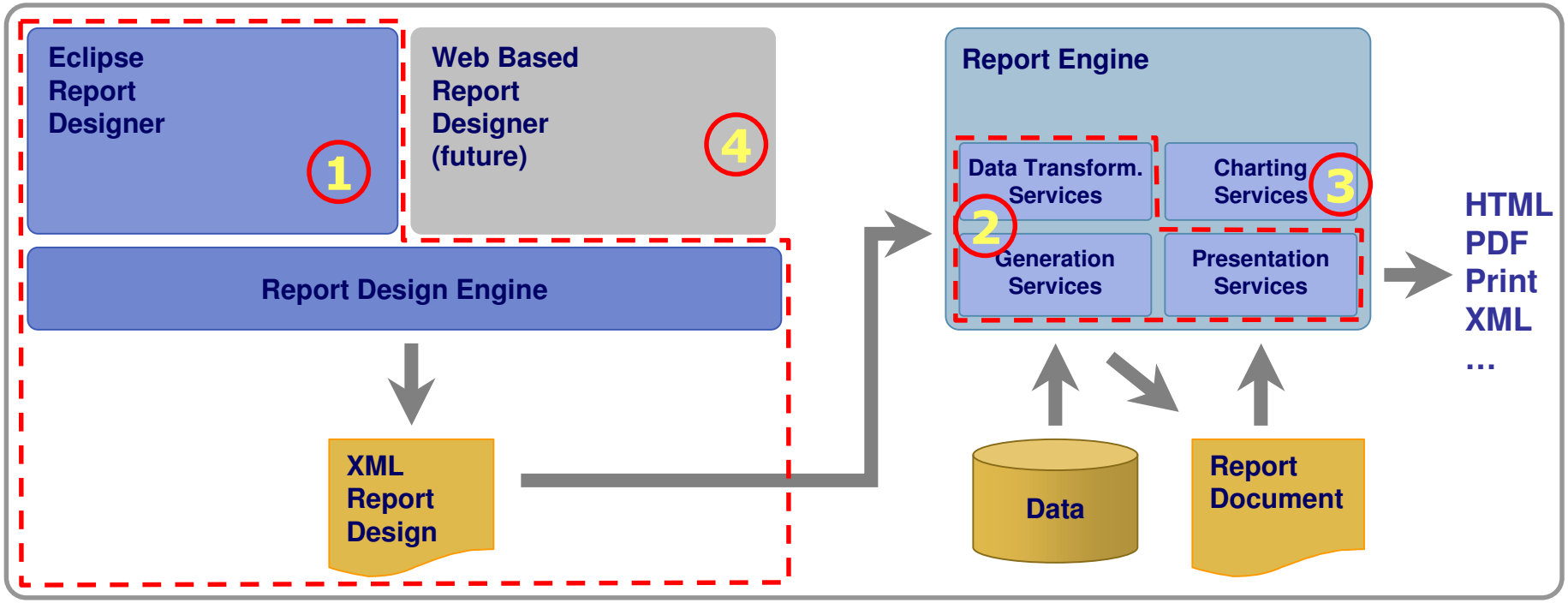
Typical Characteristics:

• Operational reports	• Simple ad hoc exploration of data	• Complex "Slice and Dice" of data
• Developer creates reports	• Business user creates reports	• Business user creates reports
• Very easy end user access	• Fairly easy to use	• More complex to use
• Highly formatted	• Typically limited formatting	• Minimal formatting
• Multiple output formats	• Interactive	• Very interactive
• No end user training needed	• Minimal training	• Requires training
• Data access can be complex	• Semantic layer hides complexity	• Semantic layer/data cubes

BIRT Initial Focus

What is the BIRT Project?

- BIRT has 4 initial projects
  - 1 Eclipse Report Designer (ERD)
  - 2 Eclipse Report Engine (ERE)
  - 3 Eclipse Charting Engine (ECE)
  - 4 Web Based Report Designer (WRD)



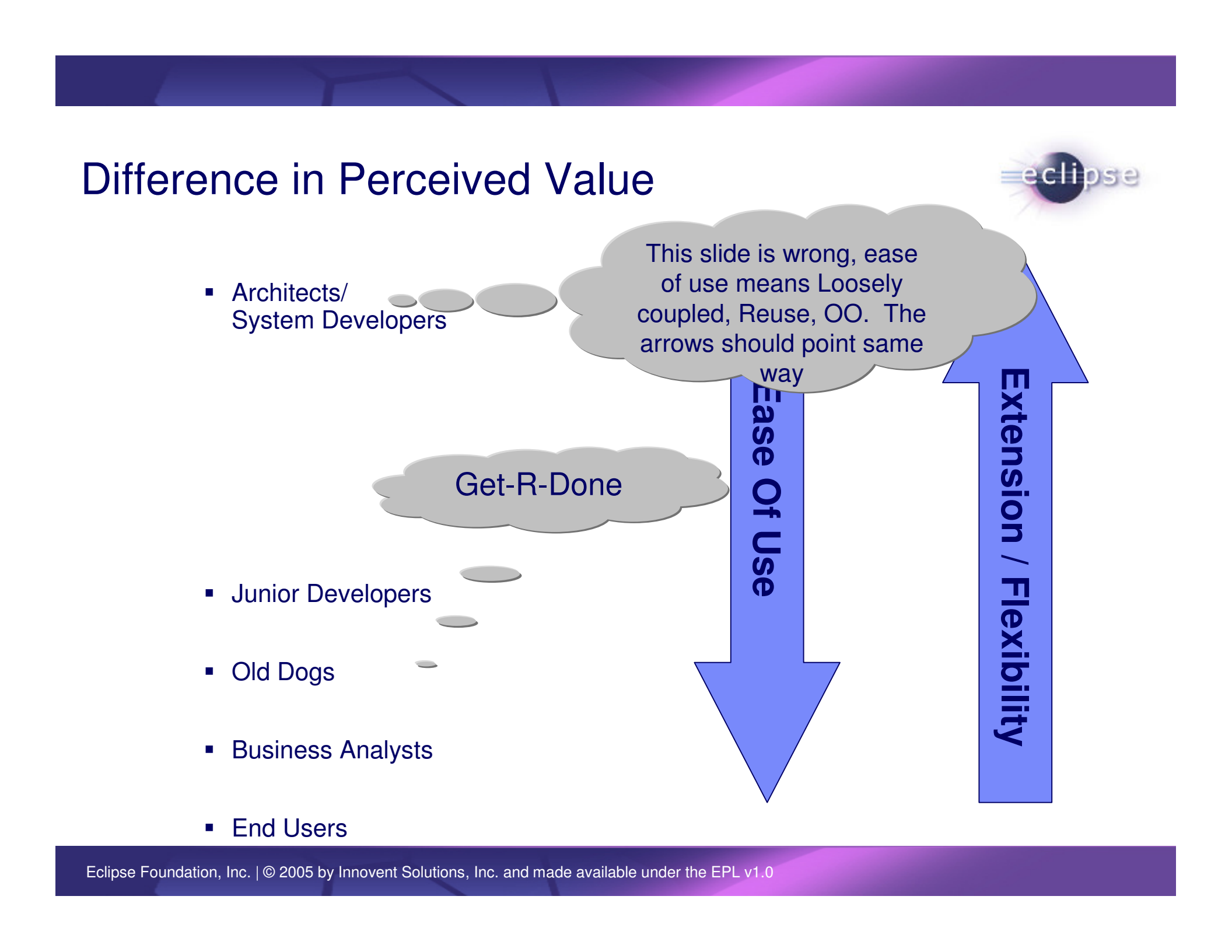
BIRT Project Goals

- **Appealing to a Broader Community**
  BIRT will address a new problem space for Eclipse that broadens the appeal of the Eclipse platform. By addressing this new market for Eclipse, BIRT aims to bring a new class of developers - report developers - into the Eclipse community.
- **Simple to Use**
  From the outset, the BIRT project regards ease-of-use to be a key objective for the project. Reporting technology is used by a broad range of users with a wide variety of skill sets - therefore the tool needs to be approachable and intuitive. The BIRT project team are focused on ensuring that users can quickly become productive with the technology.
- **Enterprise Ready**
  BIRT 1.0 will leverage the surrounding Eclipse infrastructure for Enterprise activities such as source code management and deployment of BIRT reports in the application. In addition, the initial release of BIRT will provide for support for key enterprise data bases such as Oracle and DB2.
- **Design for Extensibility: Be a Better Platform**
  Extensibility is a core principle behind the BIRT project. Extensibility will be supported in a number of areas: data access (both design time and runtime); report output formats; chart types; report components; and business logic.

About Report Developers

- Reporting Project Team Developers
  - Architects / System Developers
  - Report Developers
- Report Developers are a special breed of developer
  - Wide Range of Skill Sets
    - End Users
    - Business Analysts
    - Junior Developers
    - Experienced Developers – 'old dogs' e.g. cobol, Focus, etc.
  - Goals
    - Reports are the tail that wags the dog
      e.g. The people paying for a transactional system may never use the system, but they will expect to see reports about the system.
    - Final Product (Report) is more important than architecture
    - Good Reports create demand for more reports
  - ReUse means Cut and Paste
  - Passionate about solving Business Problems

Difference in Perceived Value



- Architects/ System Developers
- Junior Developers
- Old Dogs
- Business Analysts
- End Users

BIRT Solution – Make Everyone Happy

- Visual Report Development (End Users – Business Analysts)
  - Logical Development Model
  - Simple Palette of Drag And Drop Tools
  - Building Block Approach to complex designs
  - Customization through Property Dialogs and Component Wizards
- Report Developer Customization (Junior Developers / Old Dogs)
  - Embedded JavaScript
  - Simplified Object Model
  - Expression Builders
  - JavaScript Editor (2.0 Feature)
- Java Developer Extension (Eclipse Plugin Development)
  - BIRT Extension Points
- Java Developer Extension (API Level Develoment – Eclipse Based)
  - BIRT API

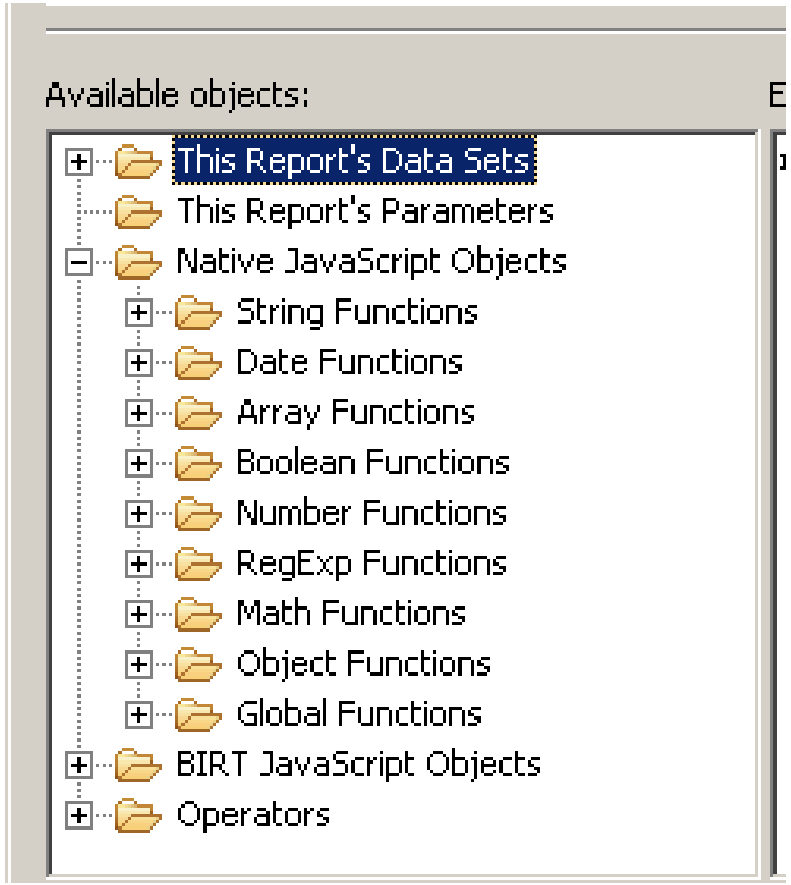
Visual Report Development - Overview

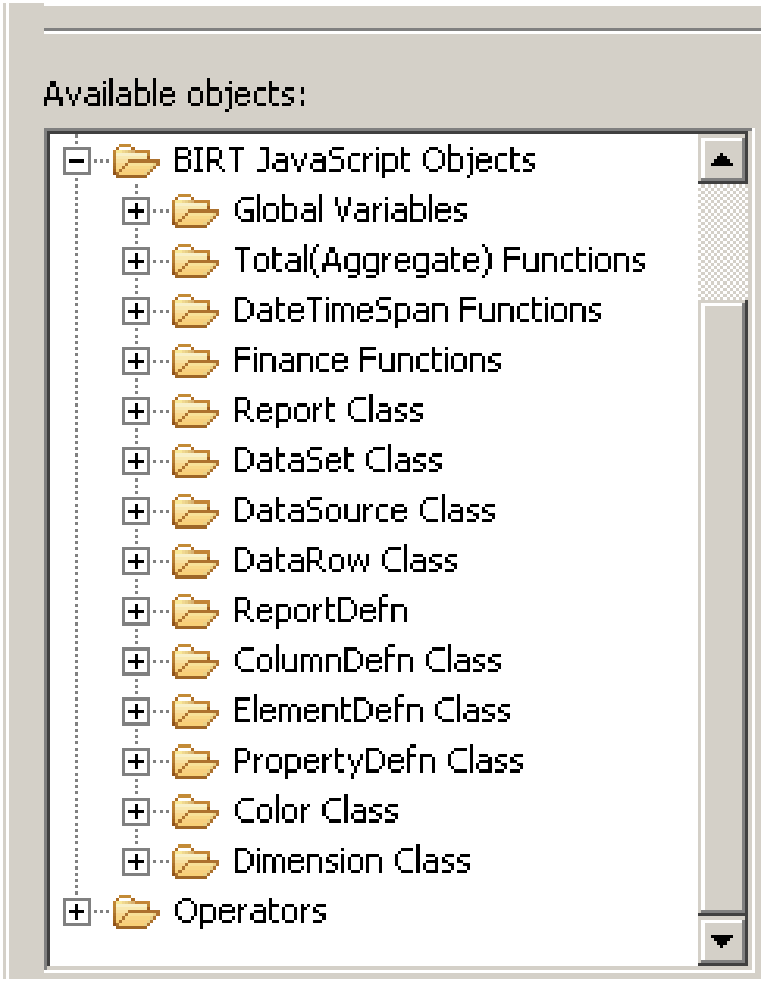
- Logical Development Process
  - Understand the Report Requirements
  - Gather Data into Rows and Columns
  - Structure the Report Items to display the data
  - Create Charts as needed
  - Format the Report
  - Deploy the Report
- Demonstration
  - Start with a Simple Report
  - Use Palette, Property Dialog, and Wizards to make more complicated
  - Deploy and Run Report in a Browser

Report Developer Customization - BIRT Scripting

- BIRT JavaScript
  - Based on Rhino
    - This is Server Side Scripting not Browser Based Scripting
  - Native JavaScript Objects and Methods
  - BIRT Custom Objects and Methods
    - Exposes the BIRT Report Object Model (ROM)
  - BIRT Globals
- Expression Scripting
  - One or More Lines of code associated with a defined BIRT Object Event.
- Method Scripting
  - One or More Lines associated with any BIRT Object Event
- POJO Scripting
  - BIRT JavaScript allows integration with Plain Old Java Objects (POJO)

JavaScript Objects





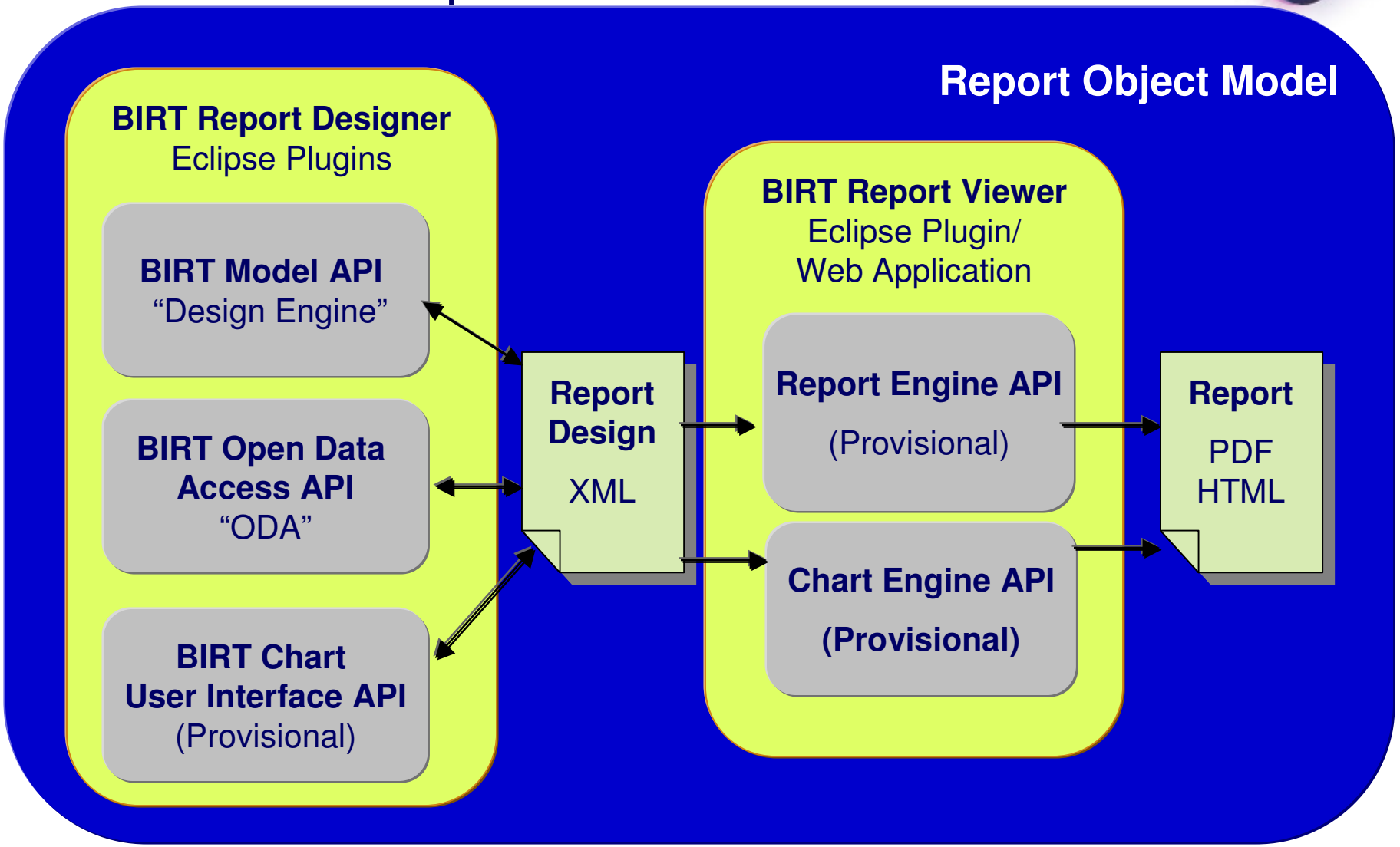
BIRT Scripting – Demonstration

- Expression Scripting
  - Modification of a Data Set Through Scripting
  - Customized Control Behavior
- Method Scripting
  - Customized Data Source Connectivity
  - Customized Parameter Handling
  - Customized Control Behavior
- POJO Scripting
  - Accessing Data from a POJO

Java Extension and Intgeration

- BIRT Extension Points
  - Allows developers to create new custom BIRT objects
  - Java Developers create new components to be used by Report Developers
  - Examples
    - Custom DataSource
    - Custom Formatting
    - Custom Charting
- BIRT Integeration Framework (BIRT API)
  - Allows Java Developers to Build BIRT functionality into their apps
  - Report / Chart Development
  - Report Engine Function
- Extension to customize, API to Integrate

BIRT API Development



BIRT Report Object Model - ROM

- XML Document
  - XSD @ http://www.eclipse.org/birt/2005/design
- ROM Documentation
  - http://eclipse.org/birt/ref/rom/index.html
- Relatively Complex / Extendable Structure
  - 38 Elements
  - 26 Structures
  - 3 Indexes
- Typically Best Approached Through Appropriate UI
  - BIRT Model API
  - BIRT Chart User Interface API
  - Open Data Access API

ROM – Name Spaces / Primary Elements

- Name Spaces
  - Styles
  - Data Sources
  - Master Pages
  - Layout Elements
  - Parameters
- Elements

Module	Common items between designs and libraries
DesignElement	Common, internal, abstract element base
ReportDesign	Describes features of one report, not inheritable
ReportElement	Base for items that can be named or customized
ReportItem	Base for visual items
Listing	Base for all Lists and Tables
MasterPage	Base for all pages

ROM Access - BIRT Model API

- aka Design Engine Api
- Builds Report Designs
- The Largest Programming Interface
  - 93 Classes
  - 2 Interfaces
- Performs a number of low level tasks
  - Read and write design files.
  - Maintain the command history for undo/redo.
  - Provide a rich semantic representation of the report design.
  - Provide *meta-data* about the Report Object Model.
  - Perform property value validation.
  - Notify the application when the model changes.
- Extension Points
  - ReportItemModel
  - EncryptionHelper

BIRT Open Data Access API (ODA)

- Convert ‘Data’ Into Rows And Columns
- Why Rows and Columns?
  - Report Developers understand Rows and Columns
  - Design Elements can be optimized for Rows and Columns
- Future Direction
  - Data Tools Project
  - Data Source Pooling – Between BIRT and JDBC based apps
  - ODA Runtime Creation Wizard ?
  - Migration to the Data Tools Project (DTP)
- Extension Points
  - DataSource

BIRT Report Engine API

- Used to Create a Custom Report Generator
  - Custom Report Viewer
  - Stand Alone Engine
  - Stand Alone Web Application
  - Report View Web Service
  - RCP
- Extension Points
  - reportItemGeneration
  - reportItemPresentation
  - emitters
  - reportItemQuery

BIRT Chart Engine API

- Stand Alone Chart Generator
- Does not require the use of the Chart User Interface
- Does not require the use of the Report Engine API
- Data is provided in a static data structure
  - When Charts render in Reports, Report Engine builds static structure from the specified ODA data and creates the static structure
  - Charts running stand-alone require a data feed
    - org.eclipse.birt.chart.datafeed — Custom User Data Set Interface
    - org.eclipse.birt.chart.model.data — Interfaces defines data structure
- The Report Engine and the Chart Engine are Loosely coupled
- JavaScript events support addition of custom business logic

BIRT Chart Engine API

- Extension Points
  - modelrenderers
  - datasetprocessors
  - displayservers
  - devicerenderers
  - aggregatefunctions

BIRT Chart User Interface API

- Allows Incorporation of BIRT Chart UI in custom applications

BIRT 1.0.1 Projects & Extension Points

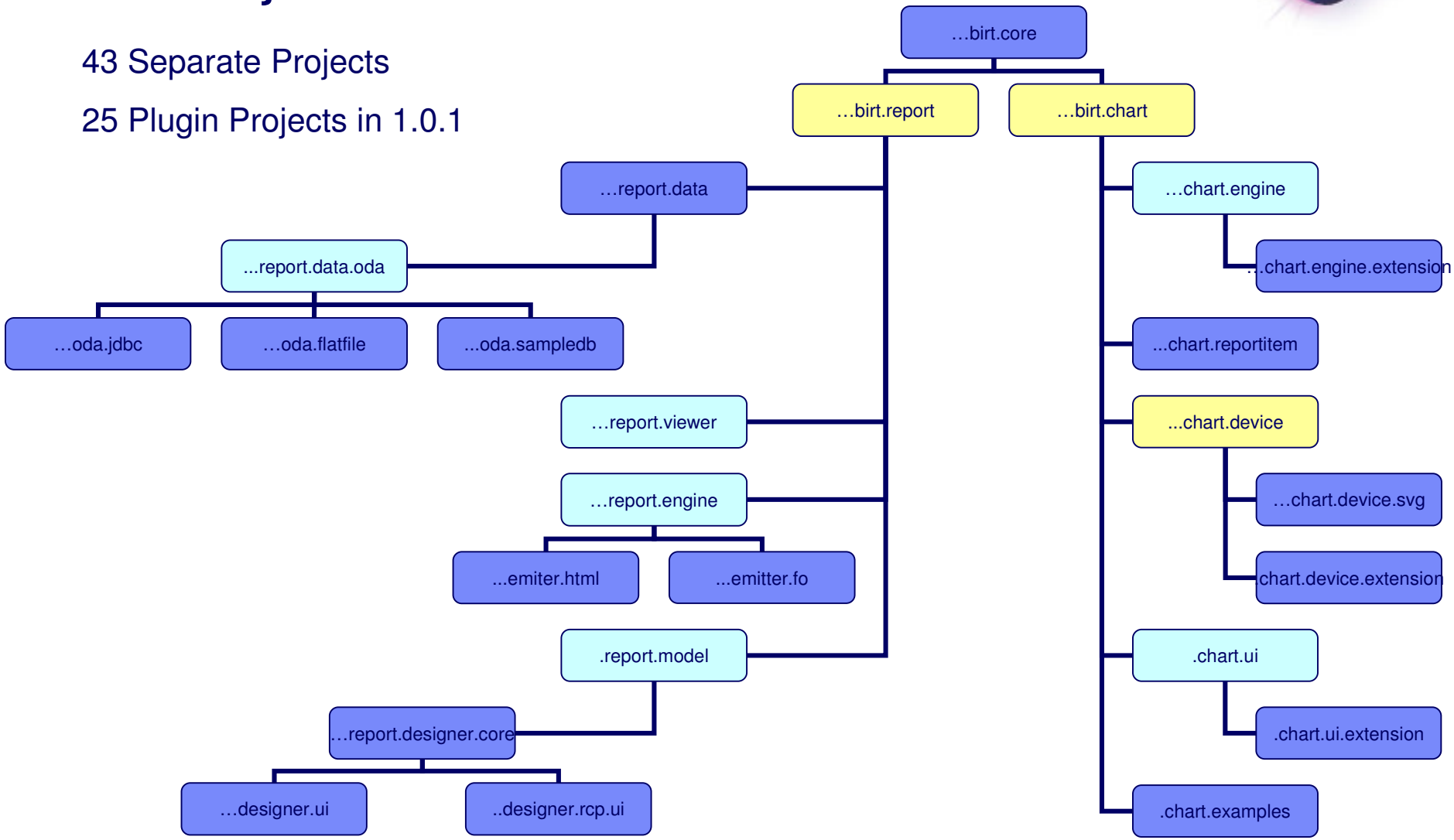
Package Name						Extension Point
org.eclipse.birt	core					
org.eclipse.birt	**report**	**model**				**reportItemModel** **encryptionHelper**
org.eclipse.birt	report	designer	core			
org.eclipse.birt	report	designer	ui			**odadatasource** **reportitemUI**
org.eclipse.birt	report	designer	rcp	ui		
org.eclipse.birt	report	data				
org.eclipse.birt	**report**	**data**	**oda**			**dataSource**
org.eclipse.birt	report	data	oda	flatfile		
org.eclipse.birt	report	data	oda	flatfile	ui	
org.eclipse.birt	report	data	oda	jdbc		
org.eclipse.birt	report	data	oda	jdbc	ui	
org.eclipse.birt	report	data	oda	sampledb		
org.eclipse.birt	report	data	oda	sampledb	ui	
org.eclipse.birt	**report**	**viewer**				**browser**
org.eclipse.birt	**report**	**engine**				**reportItemGeneration** **reportItemPresentation** **emitters** **reportItemQuery**
org.eclipse.birt	report	engine	emitter	html		
org.eclipse.birt	report	engine	emitter	fo		

Package Name				Extension Point
org.eclipse.birt	**chart**	**engine**		**modelrenderers** **datasetprocessors** **displayservers** **devicerenderers** **aggregatefunctions**
org.eclipse.birt	chart	engine	extension	
org.eclipse.birt	chart	reportitem		
org.eclipse.birt	chart	device	svg	
org.eclipse.birt	chart	device	extension	
org.eclipse.birt	**chart**	**ui**		**uisheets** **types** **changelisteners** **seriescomposites**
org.eclipse.birt	chart	ui	extension	
org.eclipse.birt	chart	examples		

BIRT Project Tree

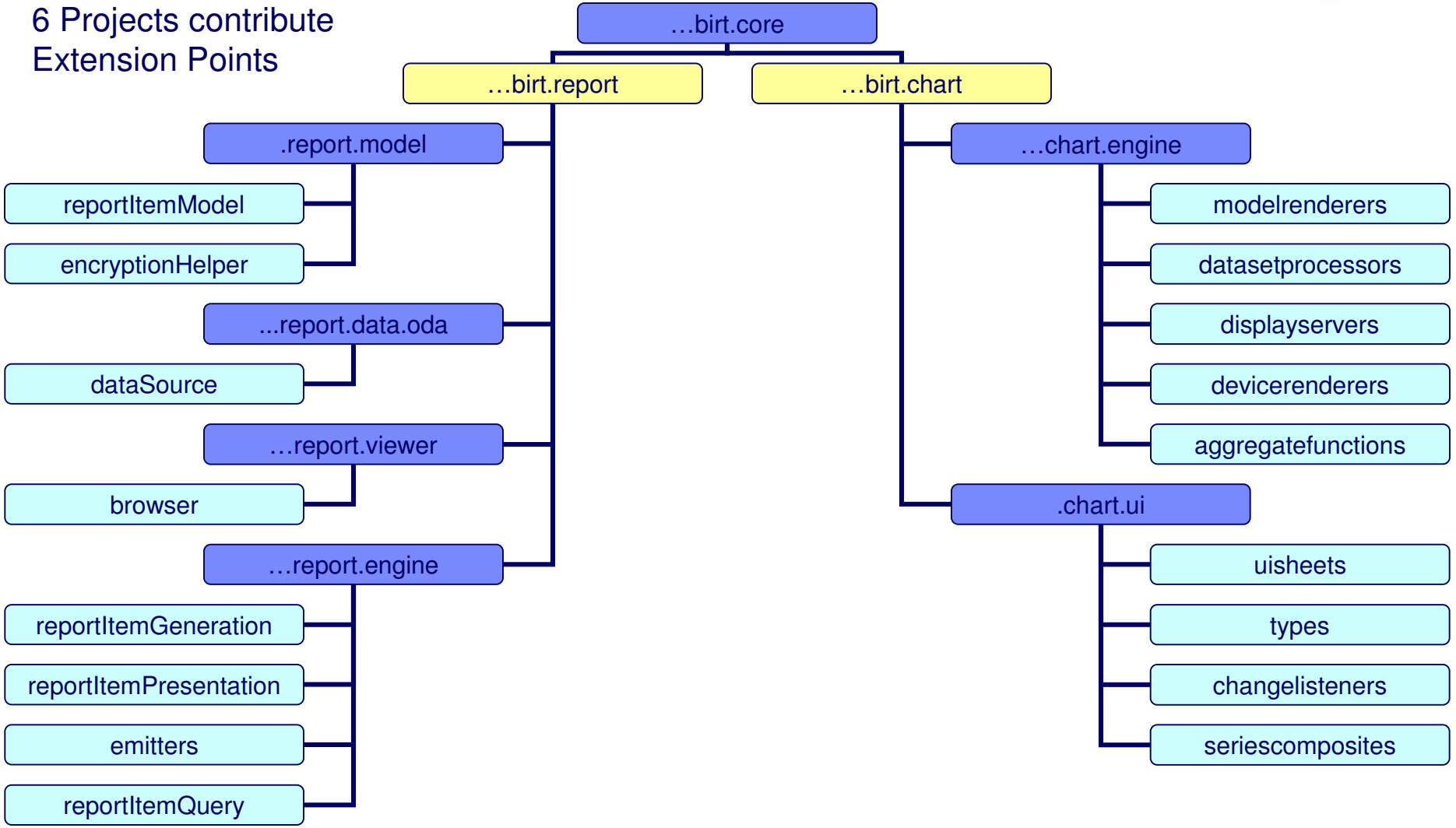
43 Separate Projects

25 Plugin Projects in 1.0.1



BIRT Extension Points

6 Projects contribute Extension Points



Demonstration – BIRT ODA

- Hibernate Data Access
- Approach 1
  - Wrap Hibernate Objects
    - Access the wrapper objects through BIRT Scripted Data Source
  - Code Intensive (not good for the visual developers)
  - Requires Understanding of BIRT and Hibernate
- Approach 2
  - Create Hibernate ODA Extension
  - More Difficult to do first development
  - Creates Re-usable objects that all developers can use
  - Data Source – Select the configuration file
  - Data Set – Select the entities to be use in the report from the config file

BIRT – Resources

- BIRT Home Page
  - Examples / Samples
  - Tutorials
  - Wiki
- BIRT Newsgroup
- BIRT Documentation
  - Eclipse Based Documentation
  - Field Guide To BIRT
  - BIRT API Reference
- BIRT JavaDocs
- BIRT Book – In Progress

BIRT - Thanks

Questions ?