Building an embedded software IDE on top of Eclipse

Gaétan Morice - Anyware Tecnologies
David Pochet - Wavecom
June 25th, 2009

Outline

Part 1 : Context

What we had to do.

Part 2 : M2M Studio

How we use Eclipse technologies.

Part 3 : Feedback

What we learned.

Part 4 : Ideas

Some though on future solutions.

Outline

Part 1 : Context

What we had to do.

Part 2 : M2M Studio

How we use Eclipse technologies.

Part 3 : Feedback

What we learned.

Part 4 : Ideas

Some though on future solutions.

Wavecom ?

- **Wavecom :**

  “Wavecom is a leading provider of embedded wireless technology for M2M communication.”

- Wavecom provide **modems**
  - ► GSM, Edge, 3G, Satellite, ...

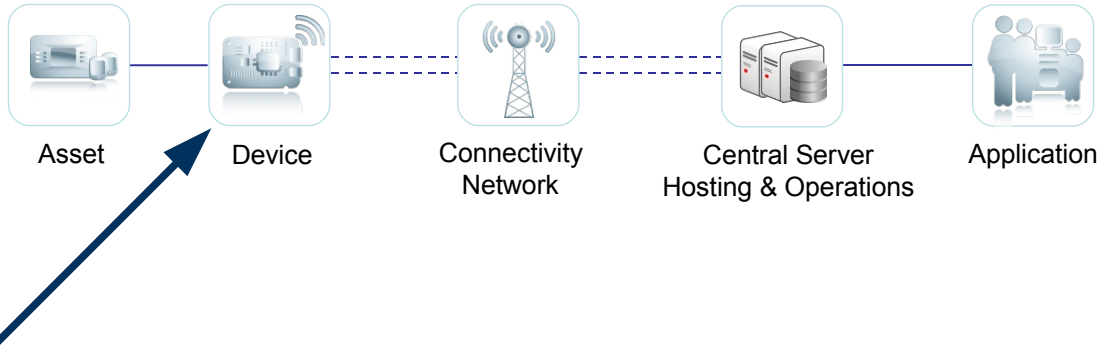
- But what exactly is **M2M** ?

M2M ?

- **Machine to Machine**

- A definition : *"M2M refers to data communication between machines."*

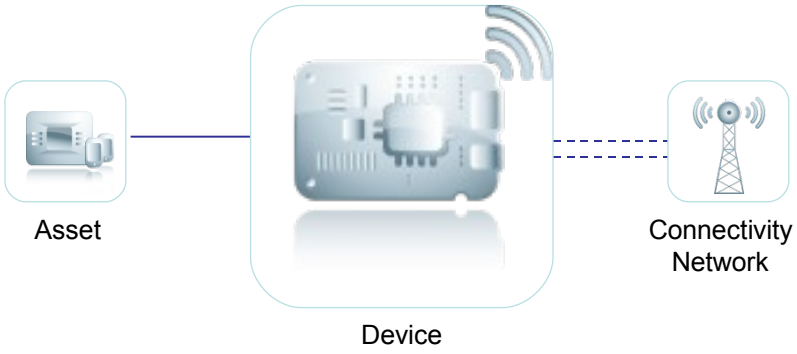
- Principle :

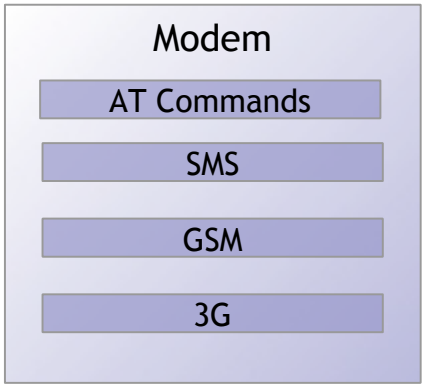


Wavecom Products

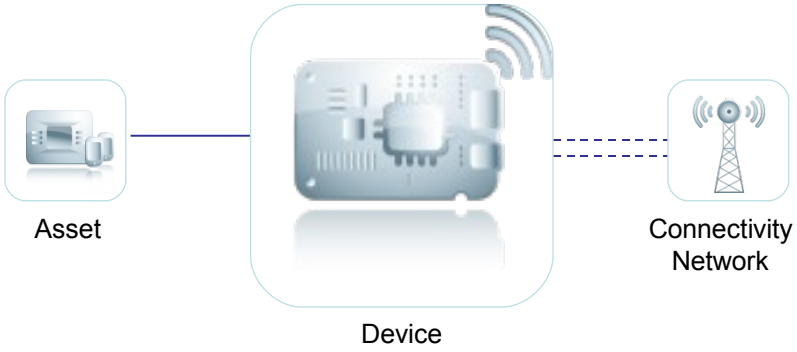
Business

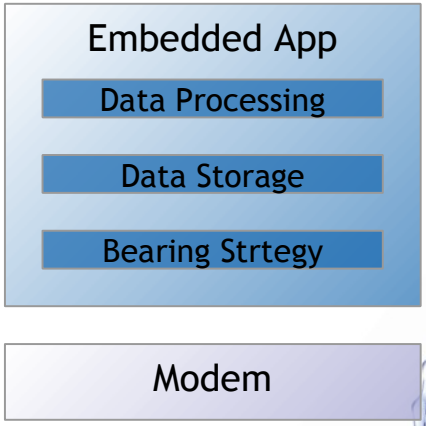
- Wavecom products may be used as **bit pipes**





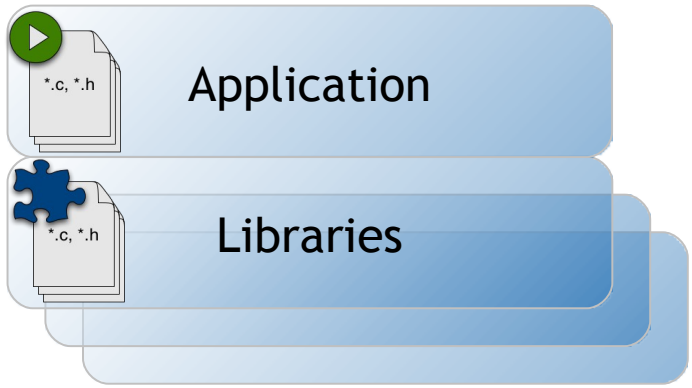
- But can also **embed** business **solutions**





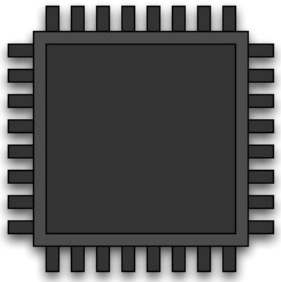
Big picture

How **embedded** solution works ?



Hardwares

- **ARM** Based
- Different families and **packagings**
  - ► WMP, Quick, Fastrack, …
- **Wireless** features
  - ► GSM, GPRS, EDGE, 3G, …
- **GPIO** for connectivity
- **Optional** features
  - ► GPS, USB, Ethernet, Bluetooth, …
- Several **memory** configuration
- Some provide **JTAG** debug

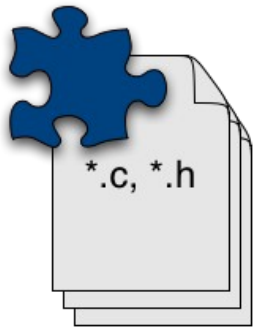


Firmwares

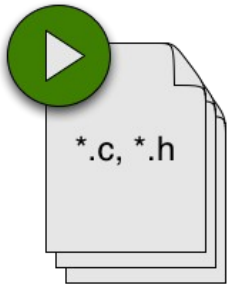
- **Raw** API for
  - Communication management
  - Memory access
  - Download & update
  - IO access
- **Services** for debug purpose
  - Traces
  - Dump
  - Remote Call
  - Memory monitoring
  - Process monitoring

Librairies

- Provided by **Wavecom**
  - C libraries
  - High level APIs
  - Optional hardware management
    - GPS, USB, …
  - Utilities
    - TCP/IP stack, encryption, …
- **Custom** library



Application



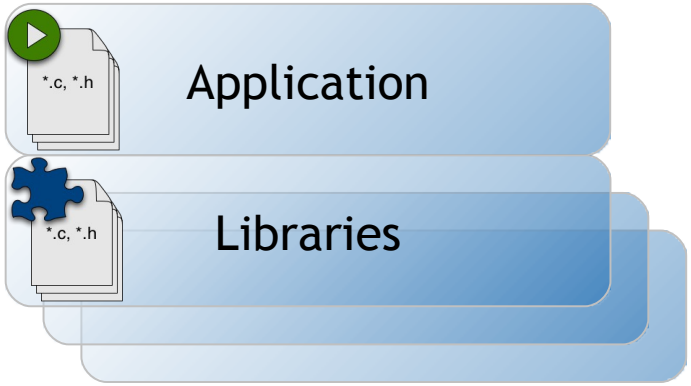
- Provided by **Wavecom**
  - ► Special Features
    - CAN Bus, …
  - ► Code samples
- **Custom** application

Tool Chains

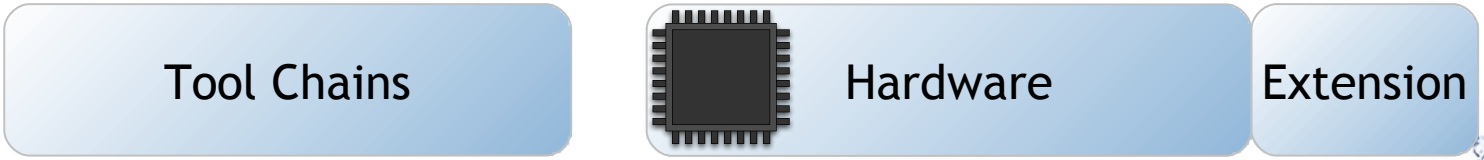
- Cross **compilation**
  - GCC
  - ARM specific compiler
  - Binaries packaging to download
- On **host**
  - MinGW

Environment

- **Complex** environment with:
  - ▶ Specific life cycles
  - ▶ Dependencies rules
  - ▶ Own documentation
  - ▶ Binaries provisioning
  - ▶ External tools set up



- An **IDE** is needed



Current tooling

- What was **provided**:
  - A bunch of makefile templates
  - Use of Cygwin
  - Some basic CDT facilities
  - Tools for download and target services

- **Problems**:
  - Install & update (several different pieces)
  - Maintain makefiles
  - Versioning of binaries
  - Newcomers
  - Synchronisation between tools (error marker, download, …)

- **A new tool was needed**

Requirements

- **Ease of Use**
  - ► User assistance in complexity management
- **Install & update**
  - ► Management of the ide, binaries, tools, …
- **Extendable**
  - ► Possibility to add new features
- **Integrated**
  - ► Code, compile, download, debug in the same tool

Outline

Part 1 : Context

What we had to do.

Part 2 : M2M Studio

How we use Eclipse technologies.

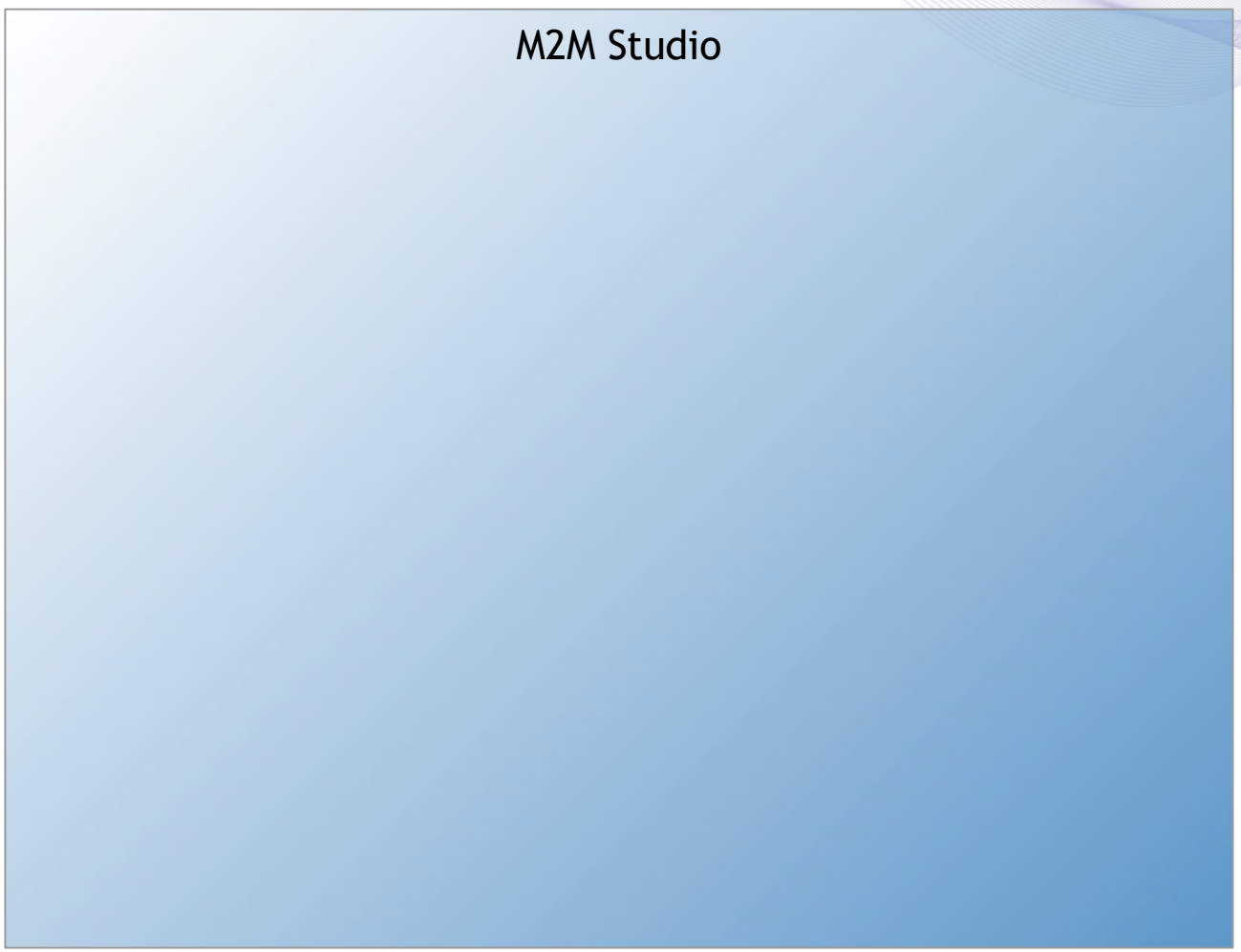
Part 3 : Feedback

What we learned.

Part 4 : Ideas

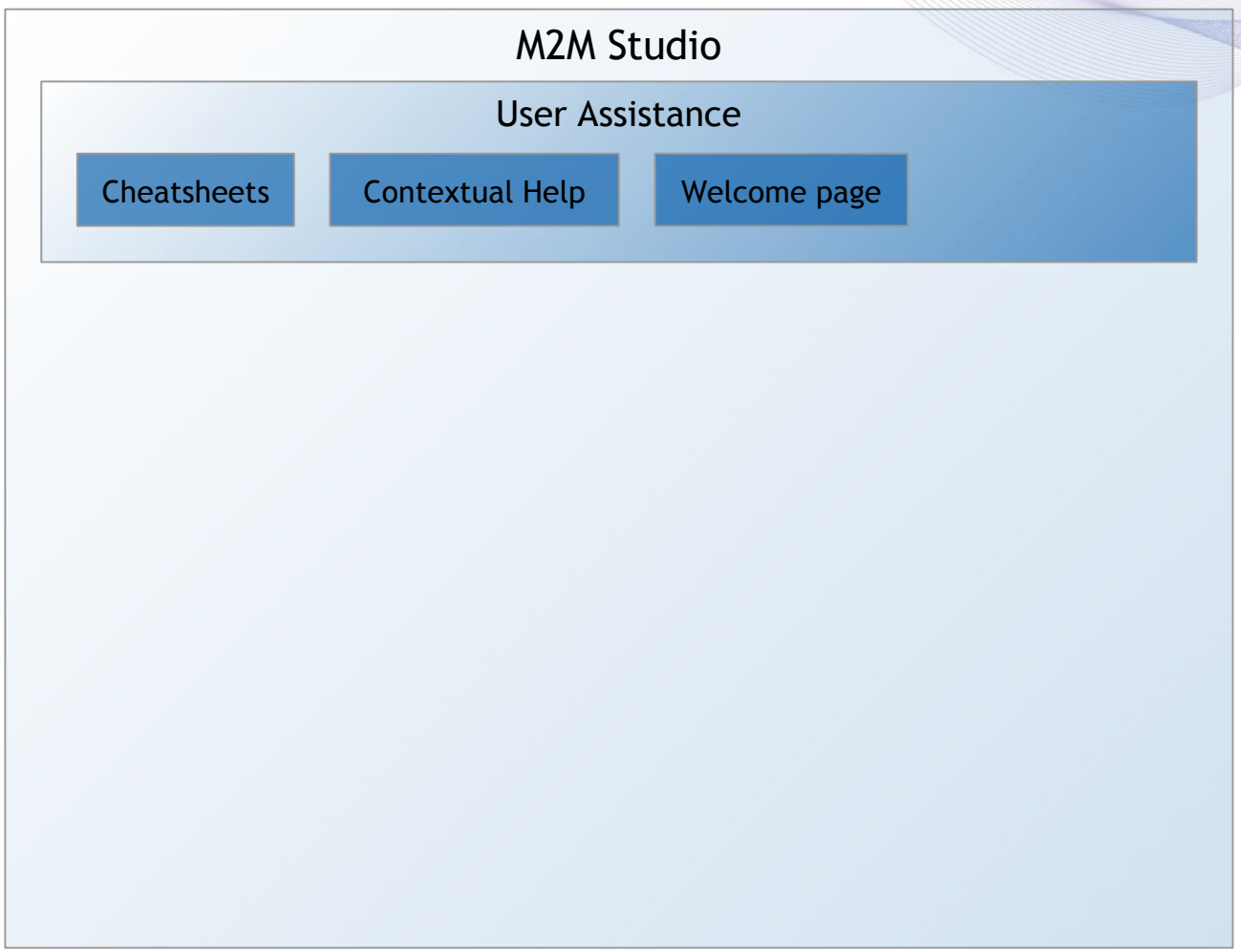
Some though on future solutions.

Need : IDE

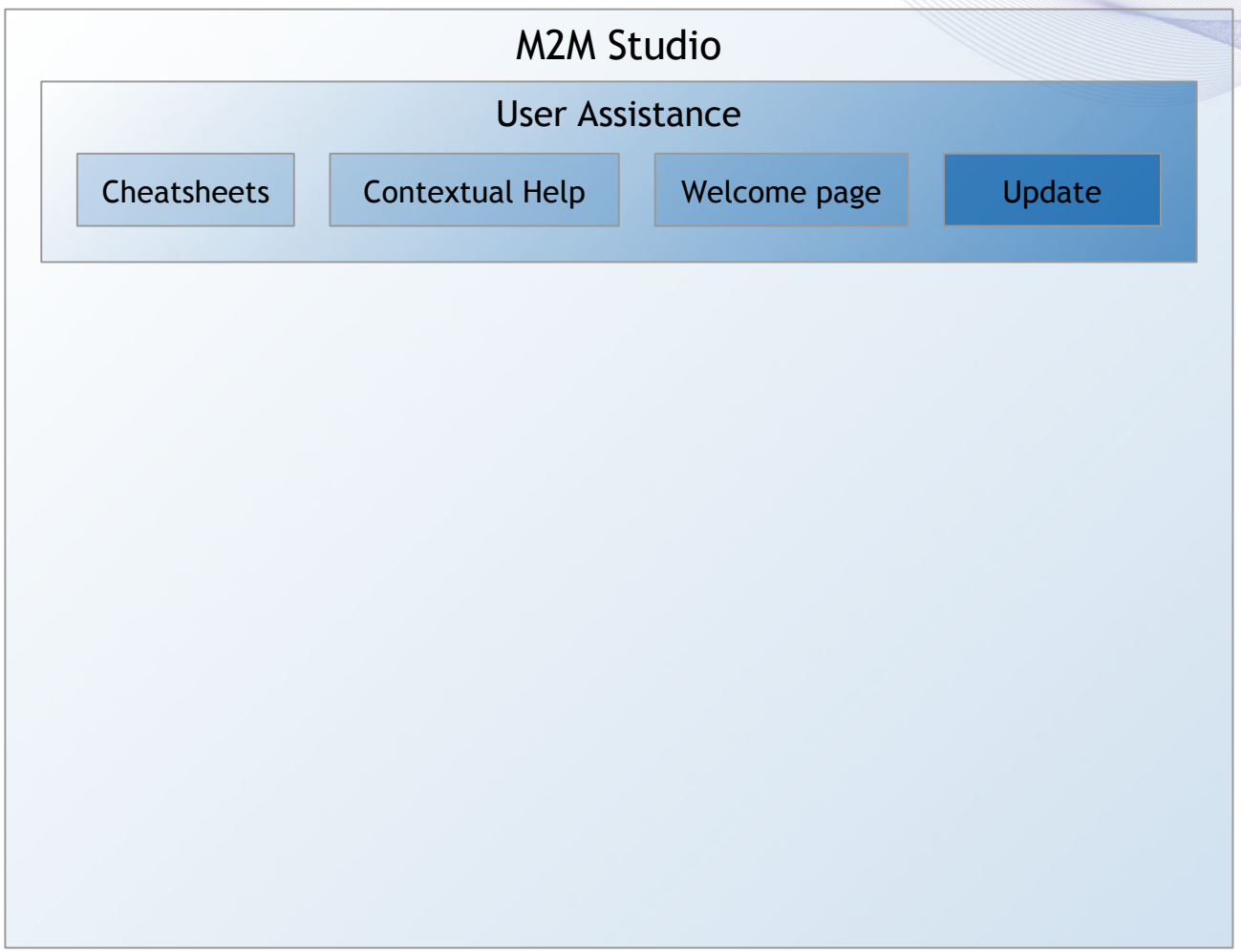


Eclipse RCP

Need : User Assistance



Need : Install / Update

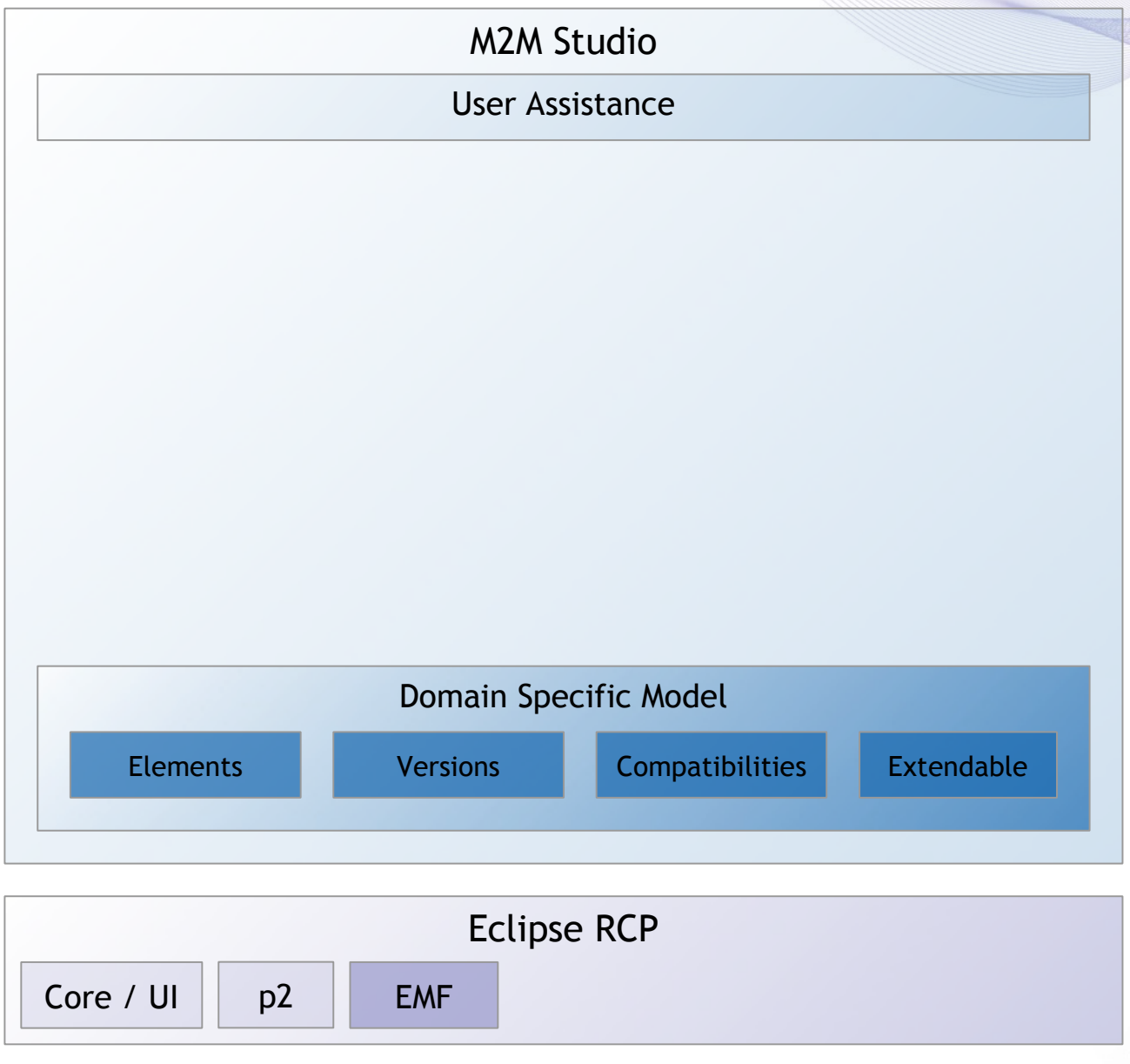


Installer

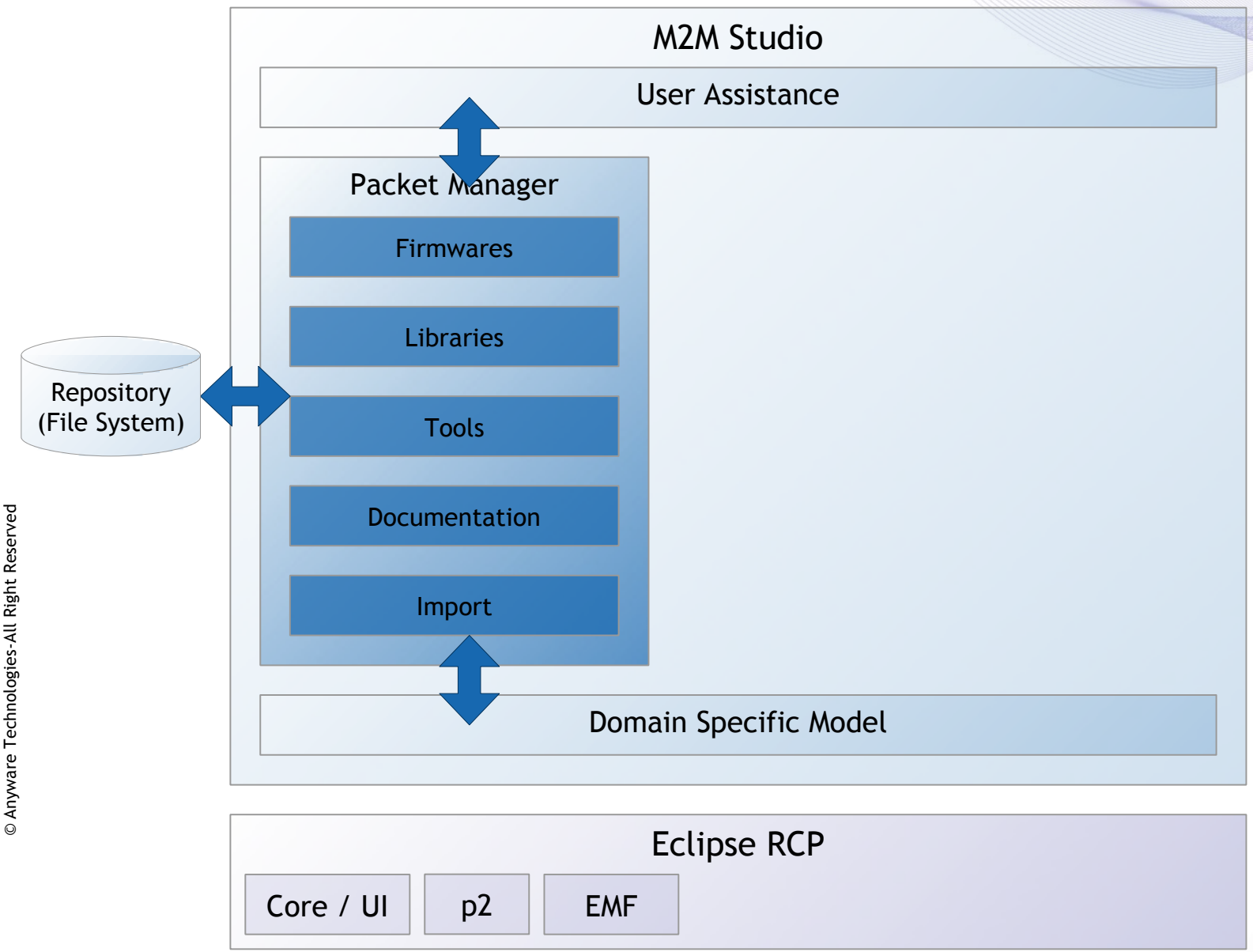
RCP

p2

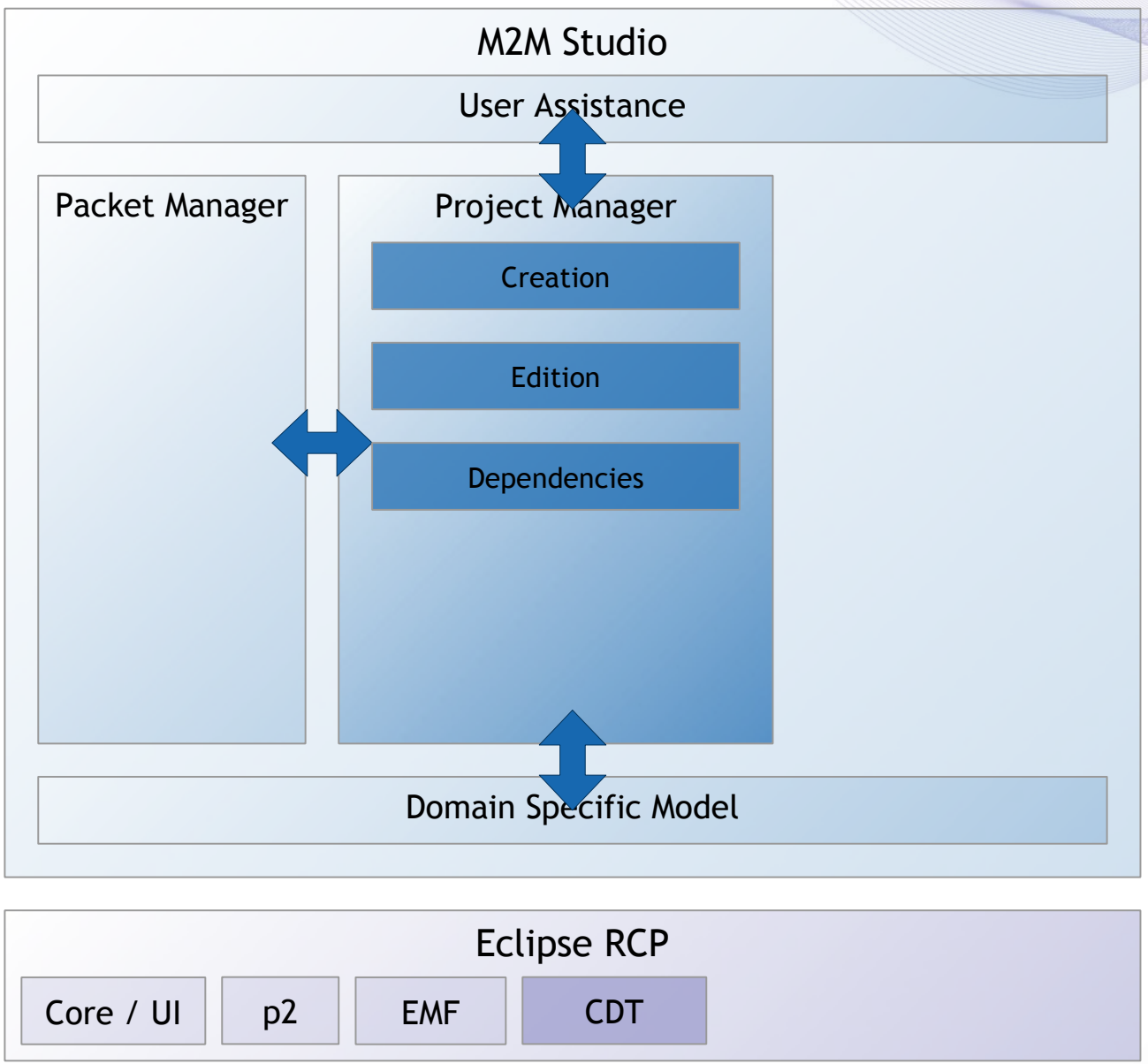
Need : Domain Specific Model



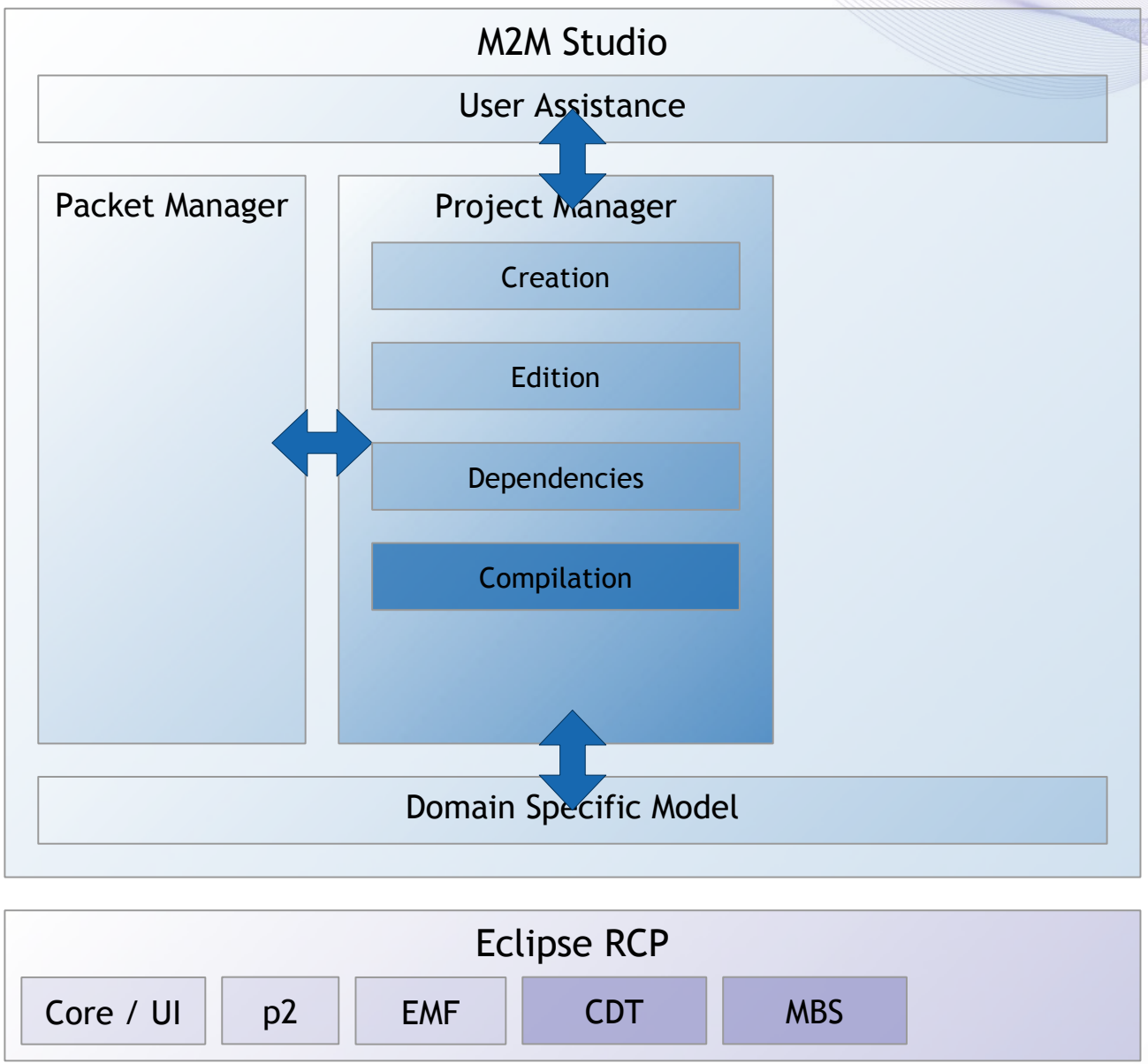
Need : Binaries Provisioning



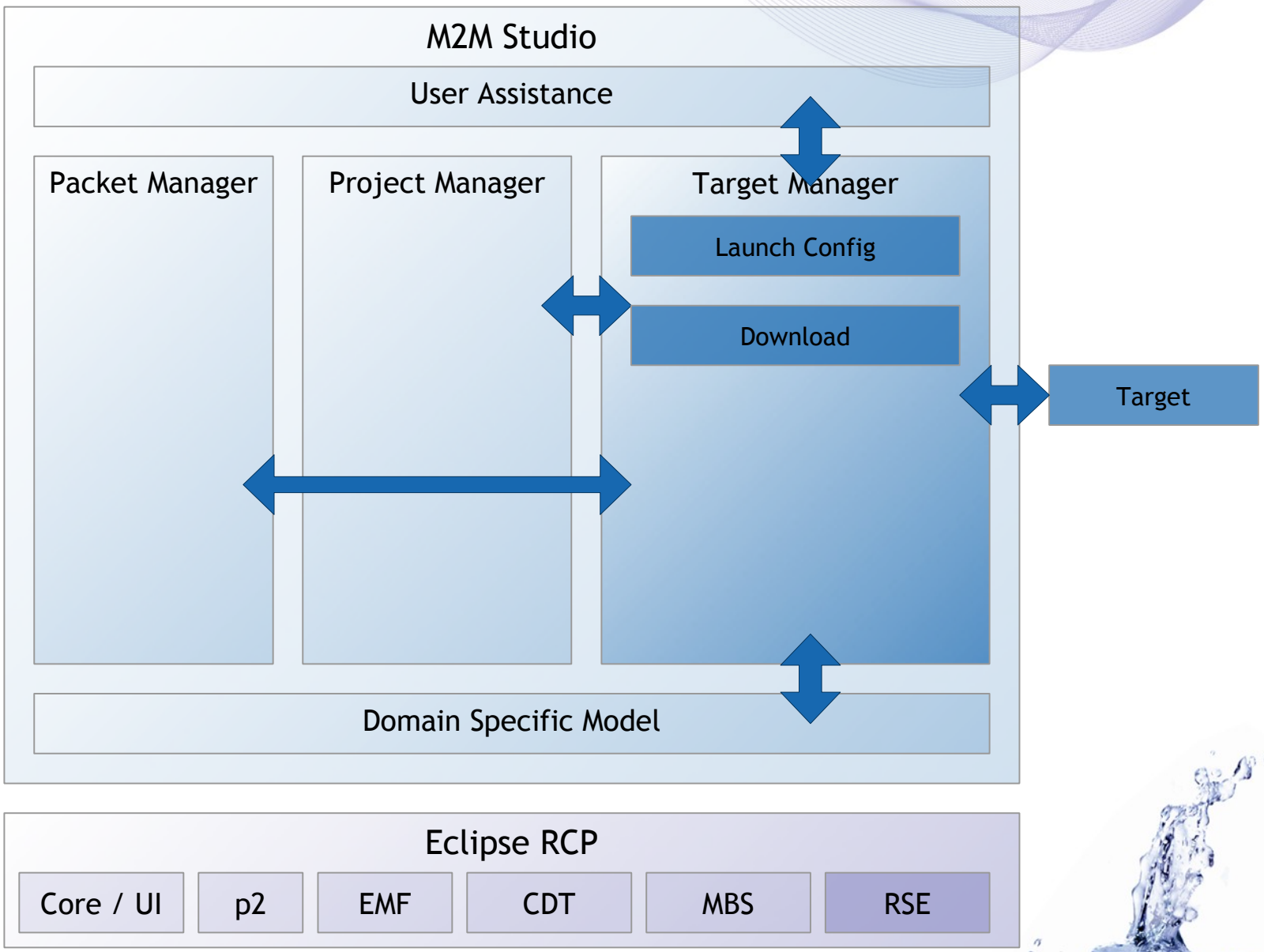
Need : Project Management



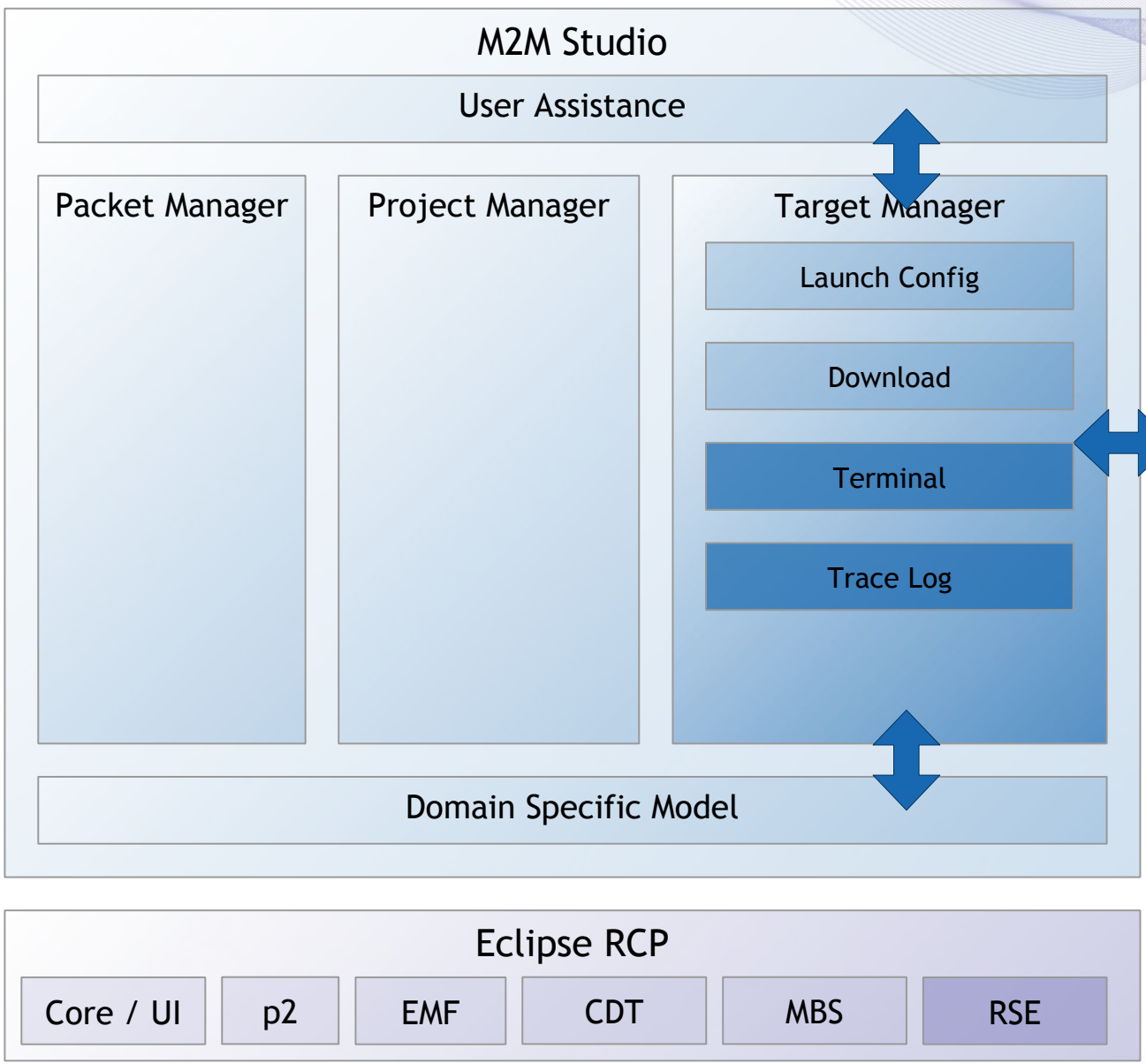
Need : Compilation



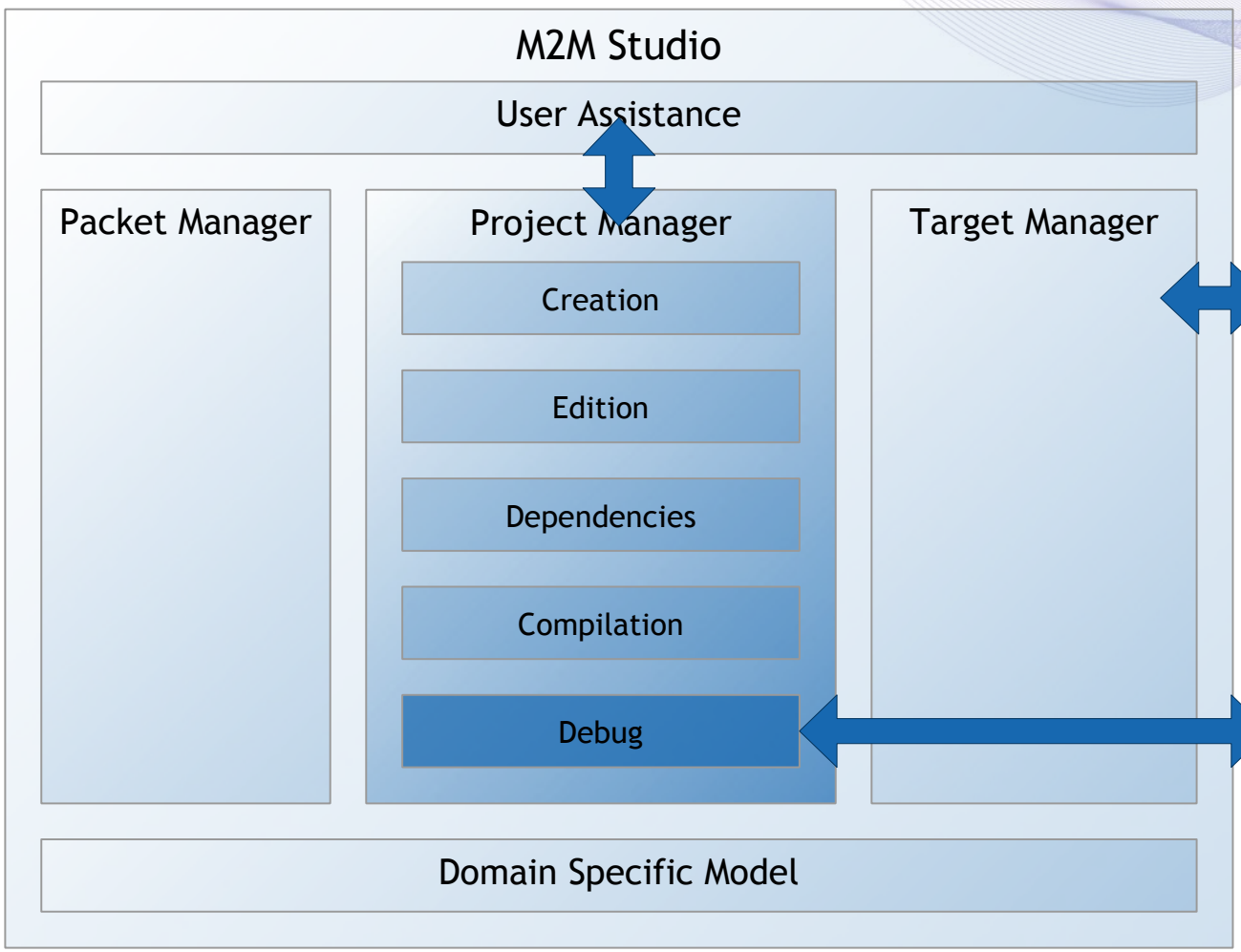
Need : Download



Need : Run



Need : Debug



Outline

Part 1 : Context

What we had to do.

Part 2 : M2M Studio

How we use Eclipse technologies.

Part 3 : Feedback

What we learned.

Part 4 : Ideas

Some though on future solutions.

PROS

Pros

- EMF domain specific modelling
  - Manage complexity
  - Re-factoring
  - Data centralisation
- CDT complete and extensible solutions
  - Configuration system
  - AST features
  - MBS possibilities
  - Debug

Pros

Pros

- RSE target and service management
  - Distinction Service / Connector
  - Model coupling
  - Ease of use
- P2 installer and update
  - Features provided

Cons

Cons

- Ergonomy
  - Set up a real ease of use policy
- Conservative user
  - Difficult to set CDT and RSE paradigms
- CDT
  - Complex project management (due to resources problems)
  - Project dependencies
  - Makefile generation
  - GUI customisation (new project wizard)
  - API documentation

Cons

Cons

- RSE
  - UI / Core de-coupling
  - TCP/IP orientation
- p2
  - Difficult to set up
  - Documentation
  - maturity

Outline

Part 1 : Context

What we had to do.

Part 2 : M2M Studio

How we use Eclipse technologies.

Part 3 : Feedback

What we learned.

Part 4 : Ideas

Some though on future solutions.

Interesting

- CDT 6 and future
  - ► DSF and TCF integration in debug
  - ► MBS and models improvement
- TCF
  - ► Embedded agent for services
  - ► Multiplexing over channel
- RTSC
  - ► Manage binaries package with lots of meta-data
- D-Pack
  - ► Packaging solutions for embedded solution development

Interesting

- OSCi Services
  - More dynamic Architecture
- E4
  - New resource system
  - New declarative GUI

anyware TECHNOLOGIES
A wavecom COMPANY
Linking Innovation to your Business

THANKS!