Polarsys

Release Engineering
[Initial & Draft Version]

Release Topics

- Roadmap
- Policies
- Continuous integration

Release Topics

- **Roadmap**
- Policies
- Continuous integration

Polarsys Release

- A release is a main value of Polarsys
- Objective for OPEES: setting up the Polarsys's release engineering, ensuring the ability to deliver Polarsys platforms
- Objective for Polarsys: delivering and promoting the first Polarsys platforms

Polarsys Release - Definitions

- A **Polarsys release** is a delivered version of the Polarsys platform
- A Polarsys release is declined in bundles
- A **bundle** contains a consistent set of integrated components.
  - Identified first: Operational(for use in operations), Experimental (for use on experiments)
  - In the future: possibility to define new bundles

Polarsys Release - Definitions

- Bundles (continuing)
  - Operational: [Definition – TBD – Integration of certified components]
  - Experimental: [Definition – TBD – Integration of validated components; some are already certified]

Polarsys Release - Definitions

- A **Polarsys release planning** defines a release number, date, a set of bundles with the components that they contain

Polarsys Roadmap

	Release1 / Date1		Release2 / Date2			Release3 / Date3			
Bundle		**Operational**		**Operational**	**Experimental**		**Operational**	**Experimental**	**[Other]**
Cmp1	Cmp1-x.y	X	Cmp1-x.y	X	X	Cmp1-x.y	X	X	X
Cmp2			Cmp2-x.y		X	Cmp2-x.y		X	X
Cmp3						Cmp3-x.y	X		X

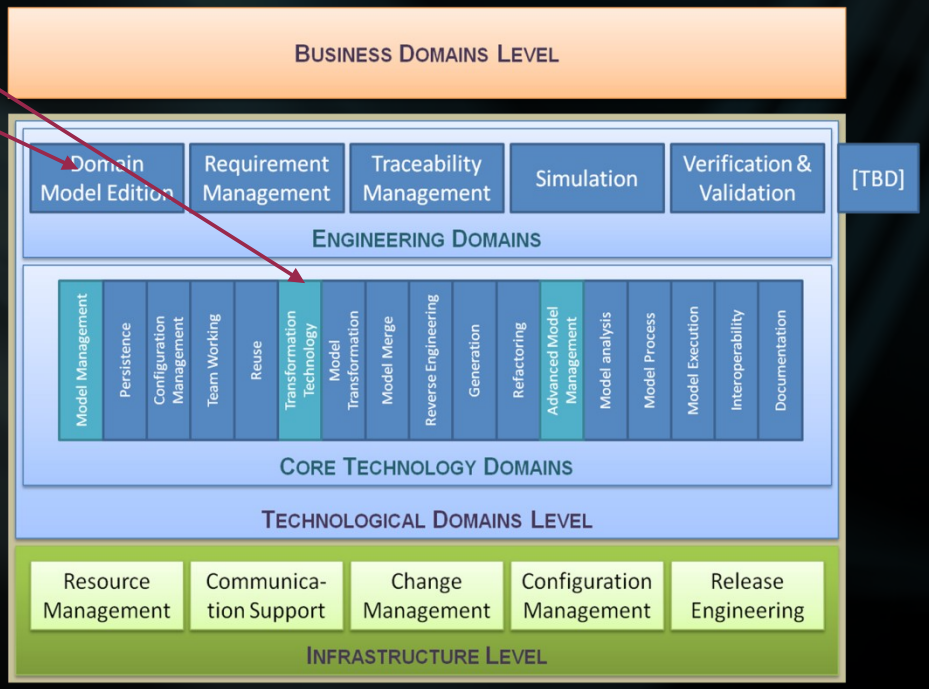
Release Target

Polarsys Roadmap

	Release1 / Date1		Release2 / Date2			Release3 / Date3			
Bundle		**Operational**		**Operational**	**Experimental**		**Operational**	**Experimental**	**[Other]**
Cmp1	Cmp1-x.y	X	Cmp1-x.y	X	X	Cmp1-x.y	X	X	X
Cmp2			Cmp2-x.y		X	Cmp2-x.y		X	X
Cmp3						Cmp3-x.y	X		X

Progressively, all the Polarsys technical domains must be covered

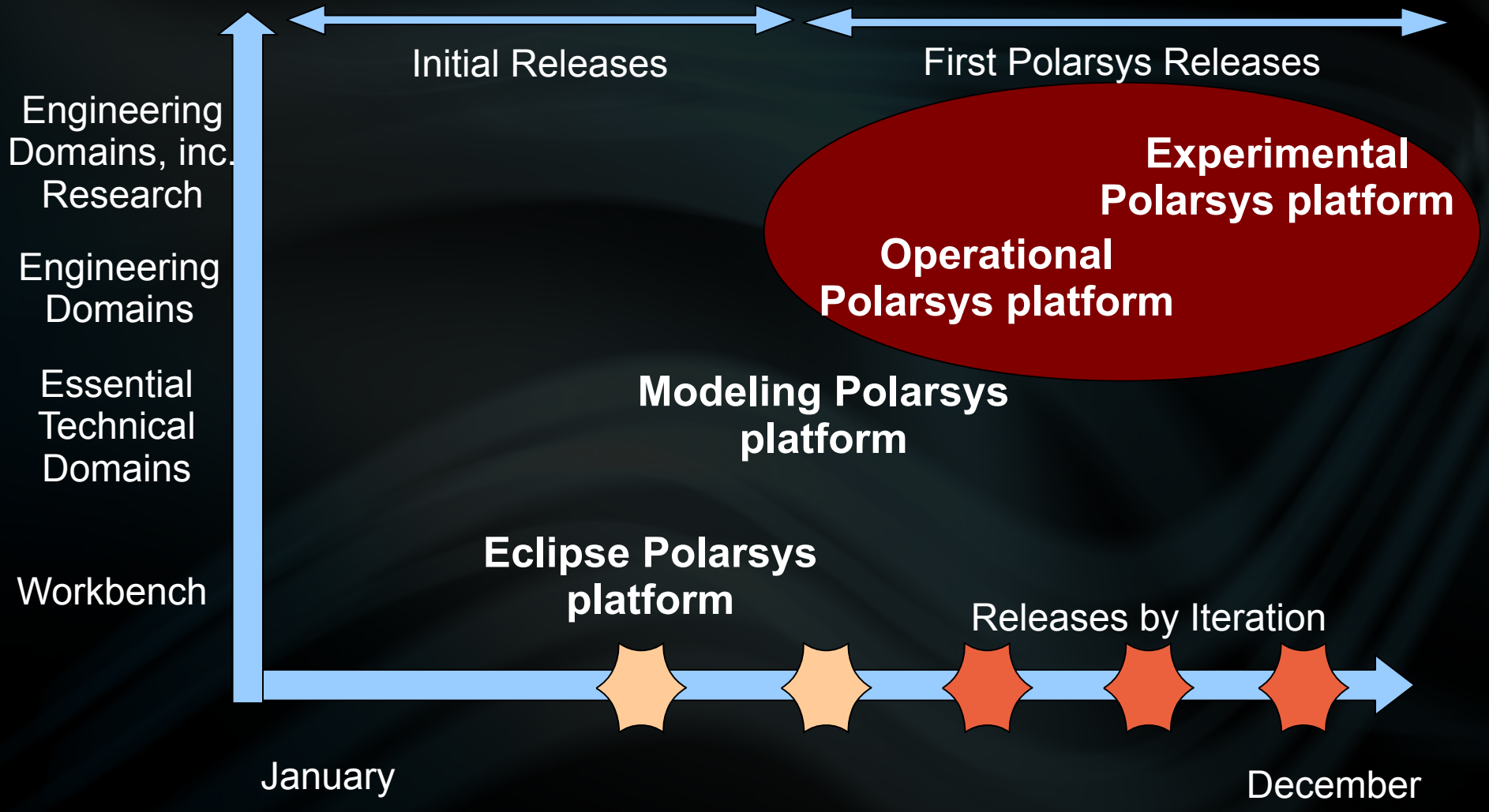
Target



Polarsys Release - Activities

- Definition of the Polarsys release planning
  - Definition of release number, date, bundles
  - Actors: Steering Committee, Architecture Board, CCB, Release Engineer
- Definition of the Polarsys release contents
  - Bundles
  - Bundles contents
  - Actors: Release Engineer, Component Leaders, Architecture Board, CCB

2012's Polarsys Releases [Draft]



2012's Polarsys Releases

- #1: Eclipse Polarsys platform
  - Set up of the release infrastructure
  - Initialization of the VLTS specific aspects (virtualization, archiving, etc.)

2012's Polarsys Releases

- #1: Eclipse Polarsys platform
- #2: Modeling Polarsys platform
  - = Eclipse Polarsys platform + Modeling Eclipse components (e.g., MDT Papyrus, Acceleo, EGF)
  - Management of several bundles definitions, builds and publications
  - Synchronization with several Eclipse projects / release train

2012's Polarsys Releases

- #1: Eclipse Polarsys platform
- #2: Modeling Polarsys platform
- #3: Operational Polarsys platform
  - = Modeling Polarsys platform + non-Eclipse components (Topcased, etc.)
  - Importation of non-Eclipse code into Polarsys (IP review, VCS, etc.)
  - Technical integration (API, etc.)

2012's Polarsys Releases

- #1: Eclipse Polarsys platform
- #2: Modeling Polarsys platform
- #3: Operational Modeling Polarsys platform
- #4: Experimental Polarsys platform
  - = Operational Polarsys platform + Non-Eclipse based bundles (gPM, FramaC, UNISIM, etc.)
  - Specific release engineering process (CAML, C, Maven, etc.), non-EPL components

Release Topics

- Roadmap
- **Policies**
- Continuous integration

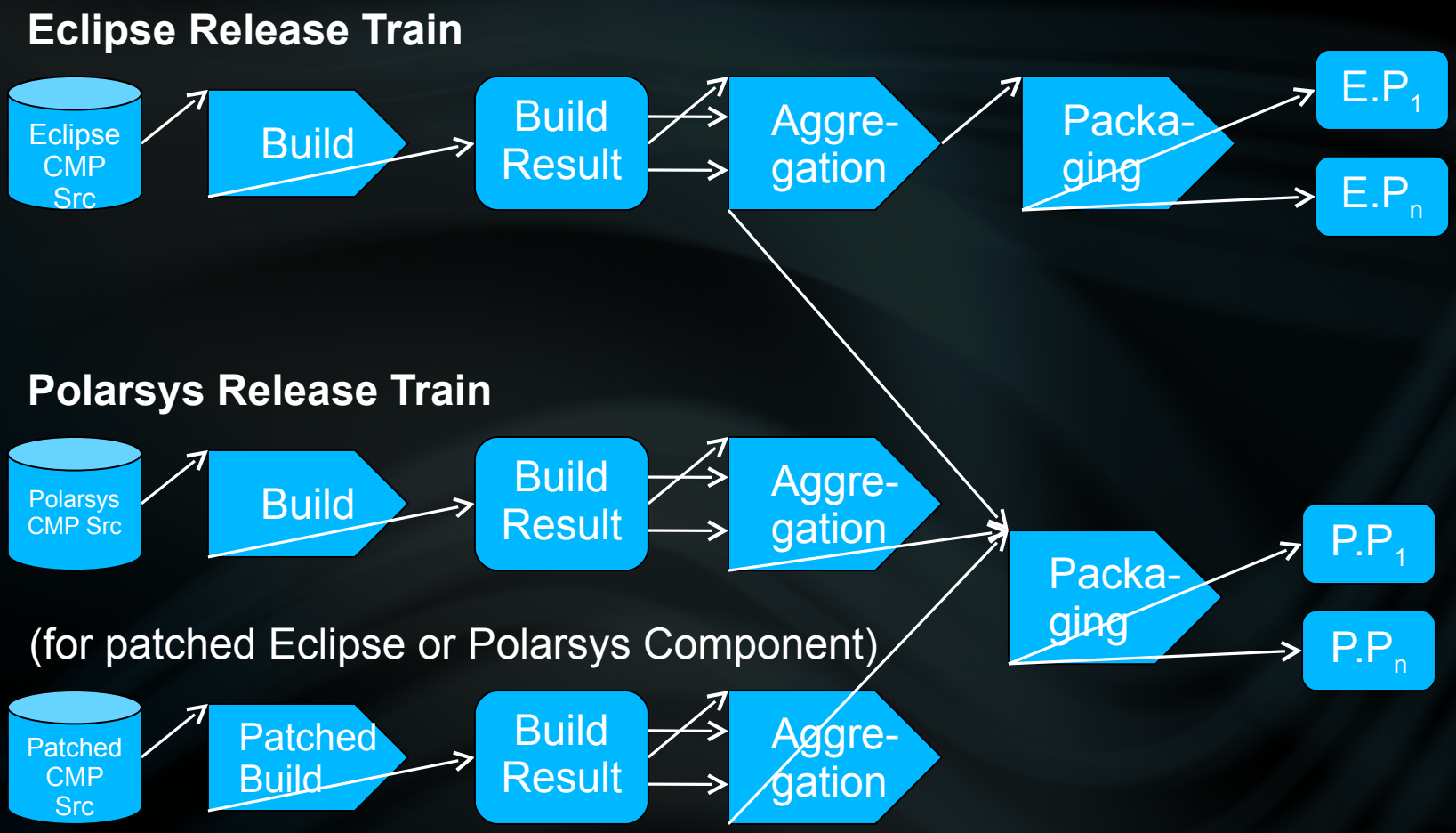
Release Policies

- The Polarsys policies define all the rules that must be respected by the component projects for a successful Polarsys release process
- Cf. Polarsys Wiki [TBD]

Release Topics

- Roadmap
- Policies
- **Continuous integration**

Release Trains



Release Engineering - Activities

- Definition of Component Build Chain
  - Actors: Component Committer
- Definition of Polarsys Build Chain
  - Actors: Release Engineer
- Control of Component Build Chain
  - Actors: Component Committer
- Control of Polarsys Build Chain
  - Actors: Release Engineer

Release Engineering Tooling

- Reusing the Eclipse environment
  - SCM (e.g., Git, SVN)
  - Tools (e.g., Buckminster, Ant, Tycho)
  - Job environment (e.g., Hudson), servers
- Possibility to use EGF for modeling all the continuation integration steps