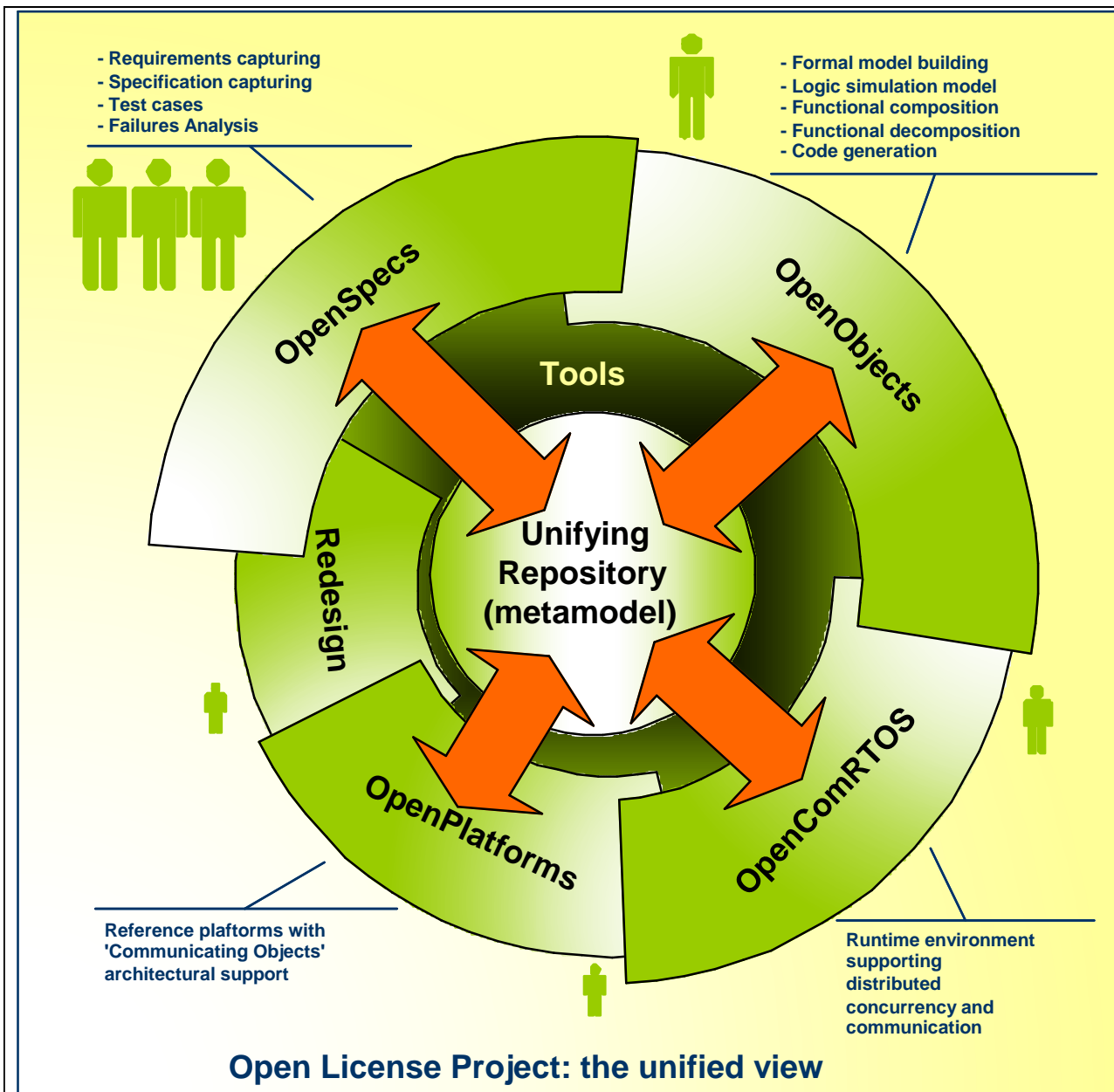


OPEN LICENSE SOCIETY

*Systematic & Unified Systems Engineering Methodologies
with Trustworthy Embedded Components*

The Open License Society researches and develops a **unified, coherent and cost-efficient engineering and development methodology for systems**. The initial focus is on embedded systems.

The key to achieving this is to use common semantics for all activities of the engineering and development process. From specification to implementation the common paradigm is based on a model of 'Communicating Objects'. This has formal roots like CSP and is reflected in the programming tools as well as the hardware. The methodology is being supported by a comprehensive toolchain. Using formal methods during the development, these tools as well as third party technology components are licensed as 'Trustworthy Embedded Components.' In this process Open License Society is building up a catalogue of products that are made available under an 'Open Licensing' scheme. The latter includes source code as well as all design, analysis and validation documents. The **Open License Society** welcomes third parties and corporate sponsors to join this unique approach to better embedded systems design.



METHODOLOGY

Systems Engineering is a human process with a lot of stakeholders. While generally seen as a technical challenge, it is also a business process with important economic ramifications. Modern insights and experience have learned us that this process should not be a linear one, but one where constant feedback at all stages is used to refine the requirements and specifications while streamlining the design and improving the implementation. This development process works with short feedback loops and short term intermediate deliverables and milestones. Often called 'Evolutionary Project Management', this is the project methodology adopted by Open License Society. This methodology is being integrated in the group-aware **OpenSpecs** requirements and specification capturing tool.

Crucial to the process of systems engineering is that common semantics are used for all activities in the process. This is facilitated by adopting a common systems view based on the paradigm of '**Communicating Objects**'. This approach is consistent with the need to separate the different concerns resulting in more orthogonality and less obnoxious side-effects resulting from overlapping functionality. This approach is also consistent with an approach aiming at 'formalised thinking'. Using **formal as well as logical model building** it helps all stakeholders to clarify the systems requirements, specifications and its architectural composition. It is part of a process in which the effort is not written but generated from verified models. The latter approach does not eliminate the need for testing, but contrary to testing, it allows to detect errors even before any implementation has started.

The Communicating Objects approach, while elegant, is also universal. It can be used to model and describe the system at most levels. E.g. at the stage of requirements building, an object will represent a sub-system. The same subsystem can then further be divided into different smaller objects that interact between them. These objects and their interactions are also the natural constituents of the formal and logical models. At the software level, each object then can be implemented using e.g. a process or task and finally mapped onto a specific hardware module.

As the runtime and its semantics are crucial in such an approach, it is the first of the tools being developed. Called **OpenComRTOS**, this is actually more of a **scalable communication layer** with real-time scheduling capabilities. Developed with the use of formal methods – which is an industries' first – it will allow to connect **MP-SoC** components, processor boards and widely distributed subsystems in a seamlessly integrated whole.

