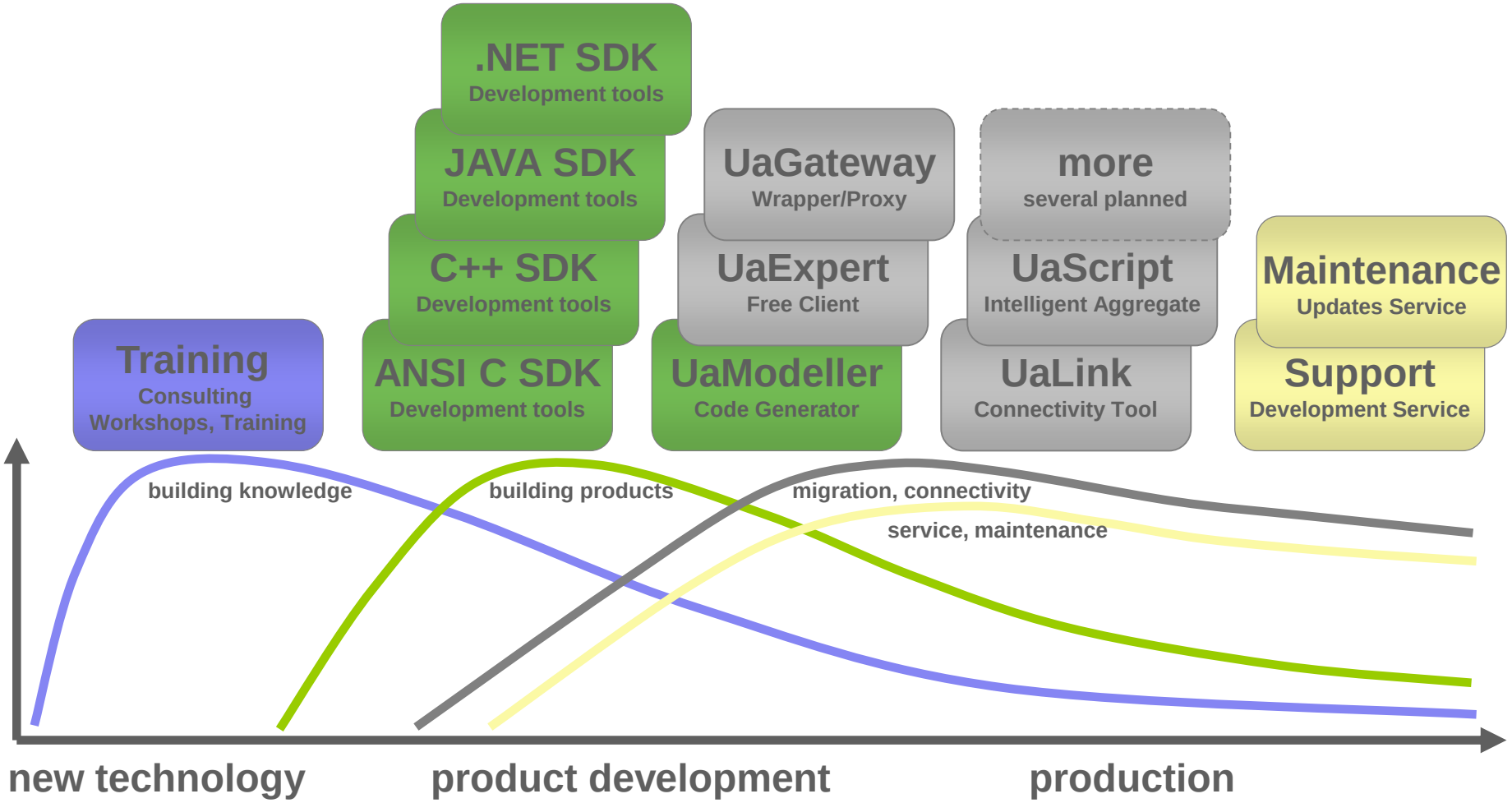
Unified Automation

OPC UA Cross Platform Tools

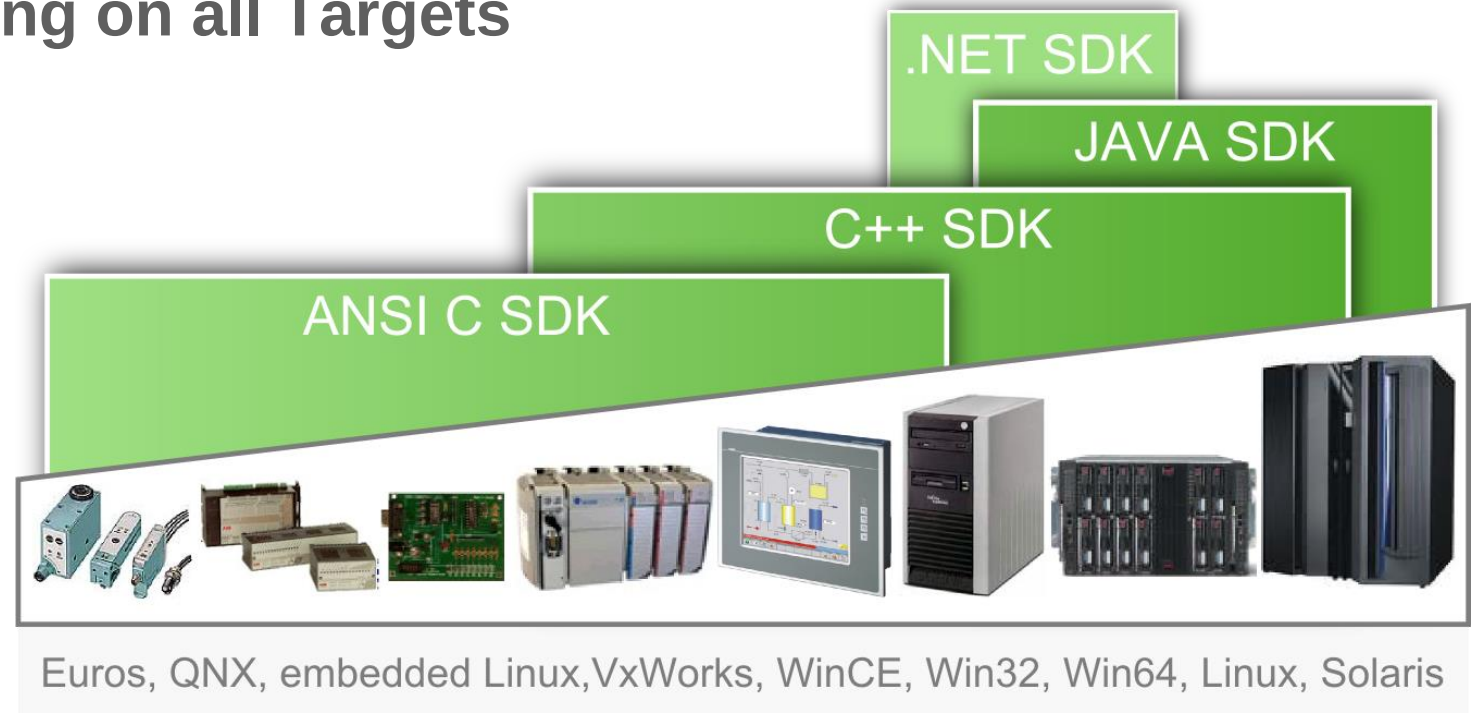
Applying OPC UA on sensor level

Market & Product Portfolio



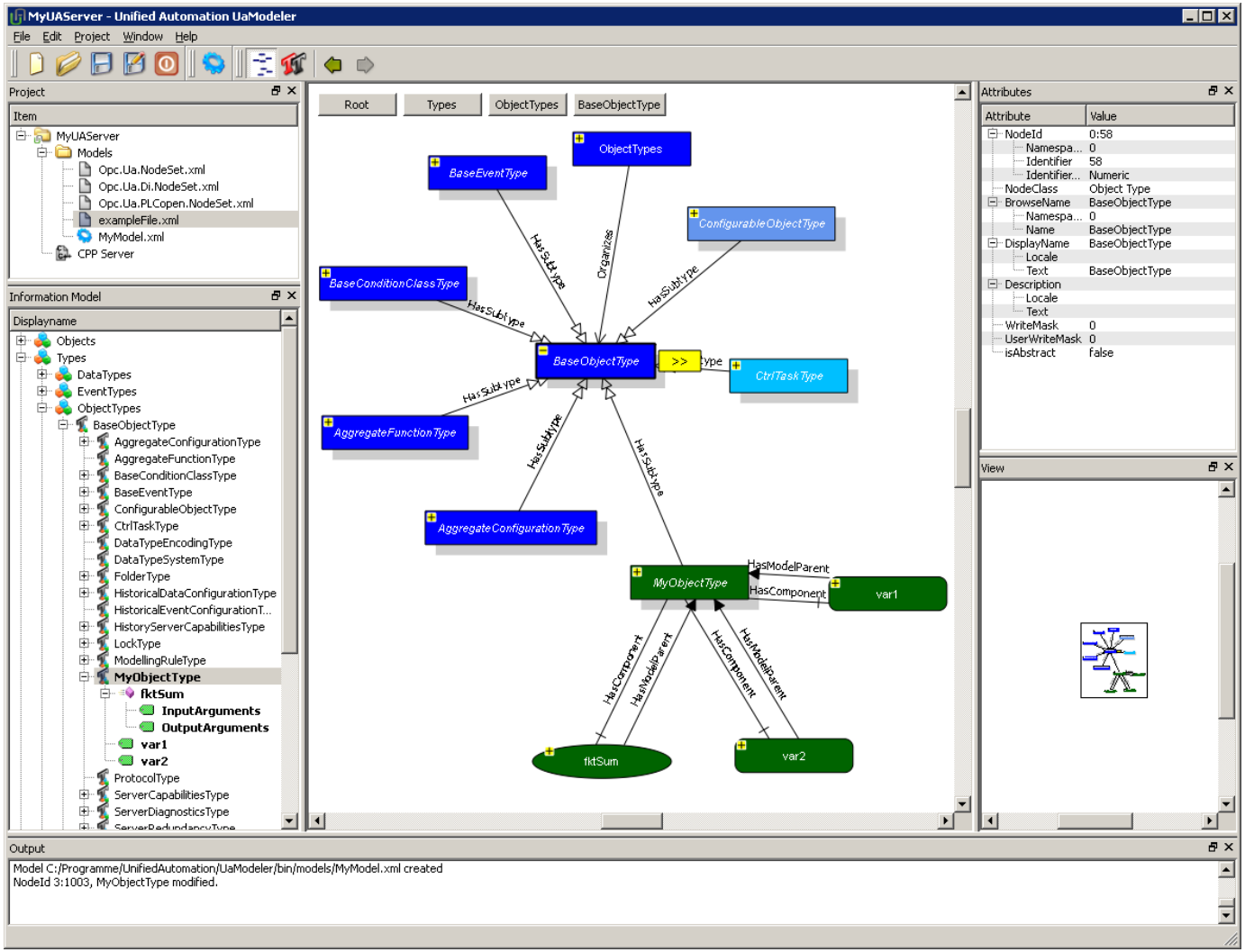
SDK Overview

- > Largest OPC UA SDK & Toolkit Vendor
- > All Languages available
- > Scaling on all Targets



UaModeler

- UaModeler turns Design into Code
- UA Nodes
- DI Nodes
- PLCopen
- YOUR Nodes



Example: Injection Moulding Machines

JAVA SDK

.NET SDK

C++ SDK

ANSI C SDK

OPC UA PubSub over TSN

C++ SDK

ANSI C SDK

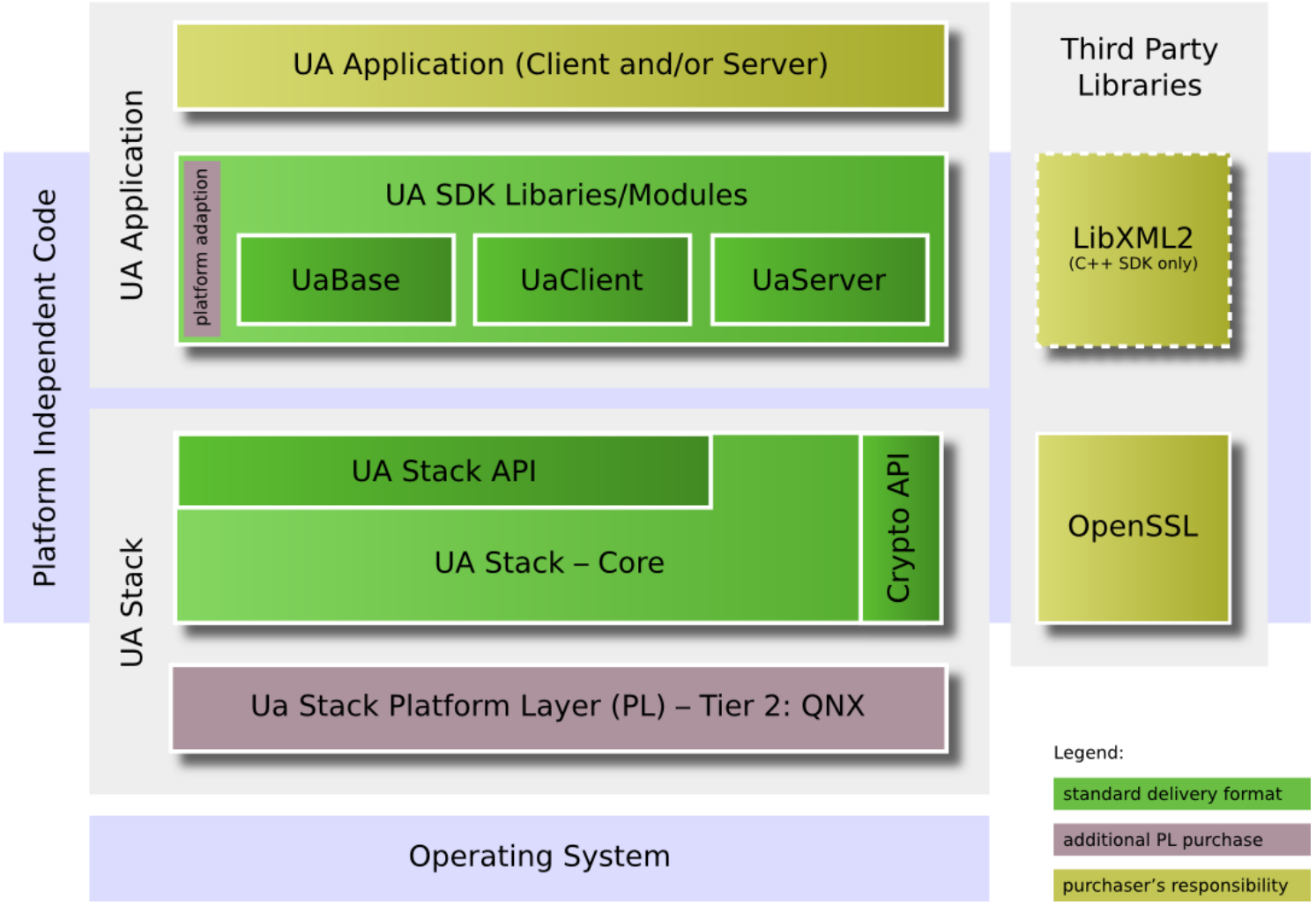
Example: Injection Moulding Machines

- **OPC UA inside the machine**
  - Up to 40 devices (controllers, sensors) with OPC UA connectivity requirements in the future e.g. RFID reader
  - Most of these devices are resource constraint
- **Requires optimized OPC UA SDK and Tooling**
- **Similar situation in Wind Turbines or Railway Systems**

Why new SDK development?

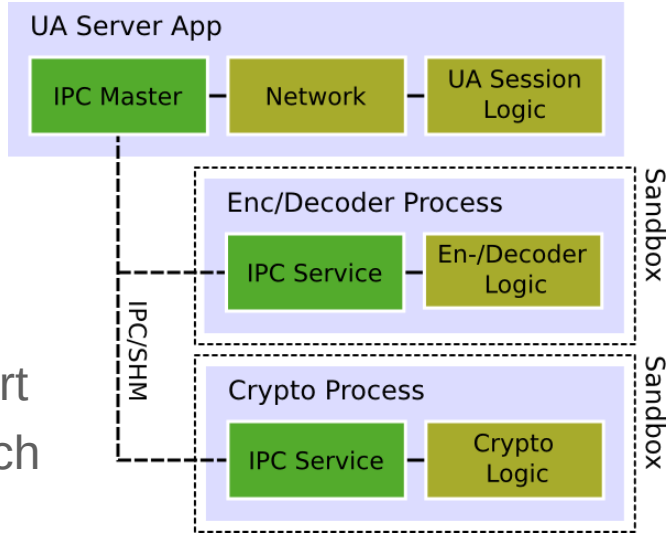
- **Analyze OPC Foundation ANSI C Stack**
  - Base for most C and C++ Toolkits / SDKs on the market
  - Not able to scale below 1 MB RAM / ROM usage
  - Significant impact to efficiency of communication
- **Goals for new small embedded SDK**
  - Scale down to 150KB or RAM / ROM usage
  - Improved communication performance
  - Reduced CPU usage
  - Designed for resource constraint devices

Technical Insight

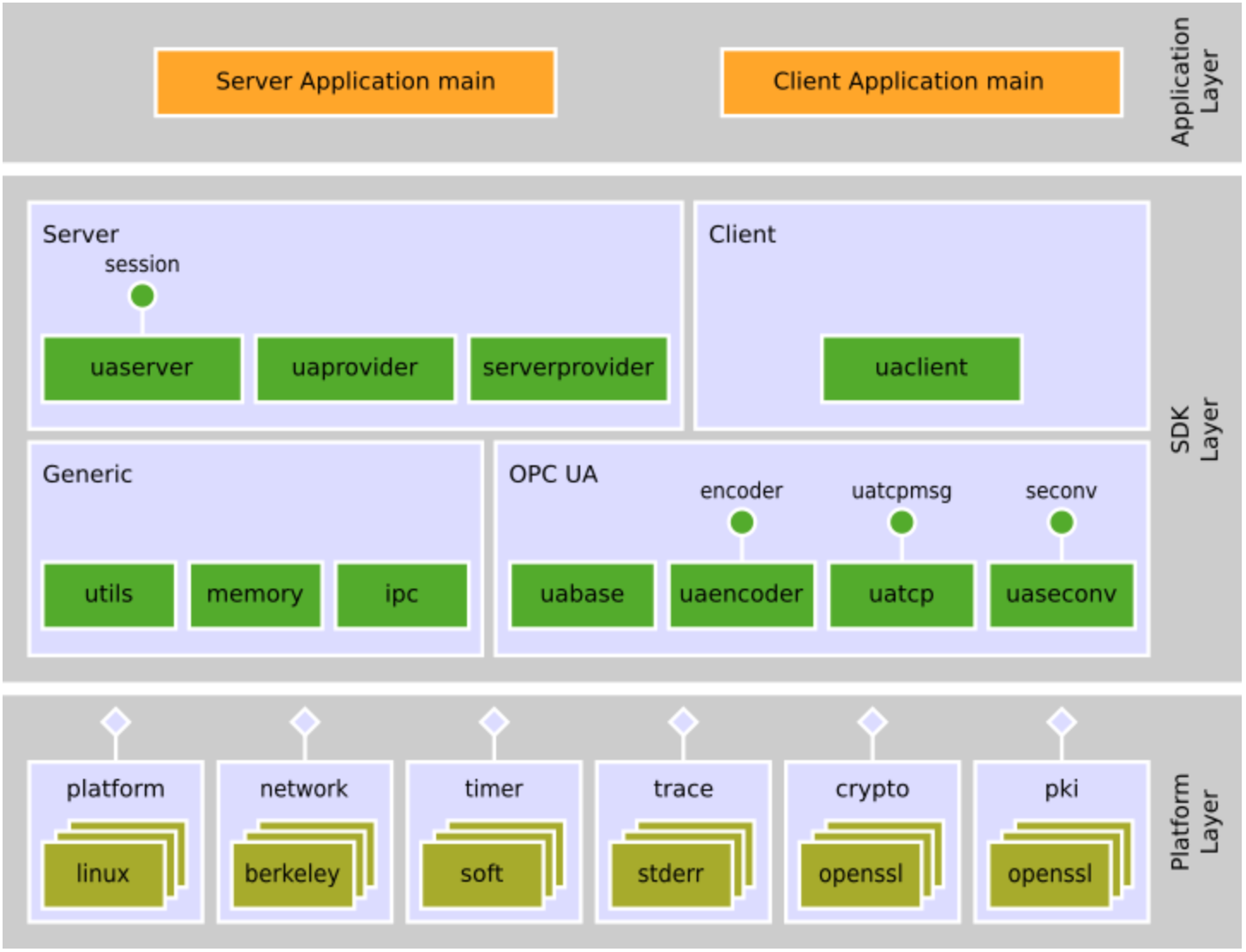


New High Performance OPC UA SDK Released

- **Key features**
  - Low memory footprint
  - Configurable memory pools
  - Address spaces can be stored in ROM to reduce RAM requirements
  - User management and authorization support
  - Includes extensive unit test framework, which helps porting the SDK to new platforms
  - Works single-threaded and asynchronous
  - Possibility to run using multiple processes to benefit from multi-core or multi-cpu systems; real parallelism can speed up the server
  - Option to run critical components like the request decoder in a sandbox
  - Security support: supports encryption and signing using OpenSSL or mbedTLS backend
  - Optional file support for loading address spaces from file (binary or XML)



Technical Insight



Applying OPC UA on sensor level

- **Key factors for OPC UA on sensor level**
  - Requirements come from condition monitoring and flexible configuration (not to replace IO communication)
  - Converged networks for real-time and non real-time (TSN)
  - OPC UA implementations that fit the requirements
  - Availability of OPC UA PubSub
- **We drive OPC UA PubSub**
  - Editor position for OPC UA PubSub
  - Leading PubSub prototyping
  - Most of the prototyping is done based on C++ SDK