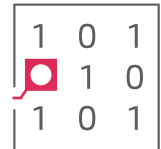


WhizniumSBE and WhizniumDBE

“Quality code for your project, generated at lightning speed”



MPSI

Your challenges implementing powerful and comprehensive Embedded and IIoT solutions

- Ever-increasing number of technologies to master: multi-threading, programmable logic, web technologies, security, ...
- Sophisticated and expensive new product developments in the context of digitalization
- Shortage of skilled staff: supply of highly qualified experts is lower than demand
- Requirement to get “digitized” products to market faster
- Unnecessary follow-up costs due to rushed project execution (“technical debt”)

please turn
for technical
details!



Our revolutionary approach:
Automated source code generation using the **WhizniumSBE/DBE**
software development tools



WhizniumSBE (Service Builder's Edition)

Innovative tool providing automated source code generation for connected embedded solutions



- (Embedded) Linux, Windows, MacOS targets
- Multi-Threaded C++ main executable
- Attached database (SQLite, MariaDB, ...)
- Multi-language HTML5 user interface
- API library for simplified integration
- Standard IIoT protocols: OPC UA, DDS
- Distributed computing
- Preferences, access control, managed file archive

Reference applications: industrial control, sensing, machine learning, artificial intelligence and many more

Your advantages



- Up to 70% time savings in the development process reduce cost and enable faster time-to-market
- Automated writing of source code for non-core - but crucial - functionality significantly enhances productivity
- Clean source code structure simplifies teamwork and maintenance
- Source code free of external references: no closed-source libraries are required and all functionality is in plain sight
- Hassle-free intermixing of automatically generated source code with manual additions
- Attractive Software-as-a-Service model, deployment options on-premise or in the cloud

WhizniumDBE (Device Builder's Edition)

Innovative tool providing automated source code generation for programmable logic projects



- FPGA and microcontroller based hardware targets
- Modular command set as VHDL state machines
- Custom host C++ device access library

Reference applications: high speed signal processing, machine vision and many more

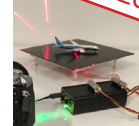
Onboarding

- Numerous sample projects
- Documentation and demo code online
- Training and consulting services
- Hands-on implementation for industrial customers

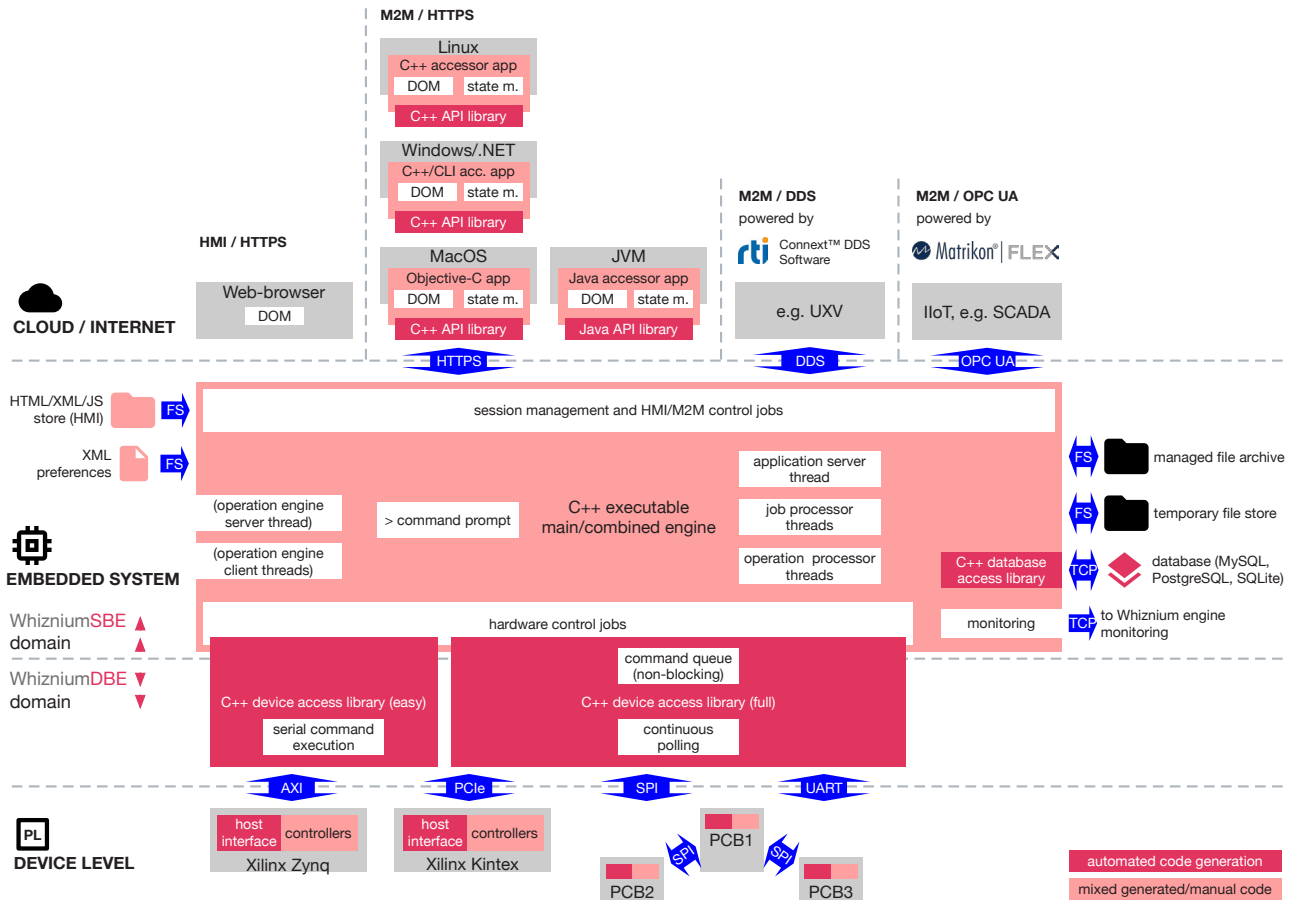
Hardware starter kit in three flavors workstation | embedded | FPGA

- Tabletop 3D laser scanner
- Toradex / Digilent boards
- Test license and WhizniumSBE/DBE projects

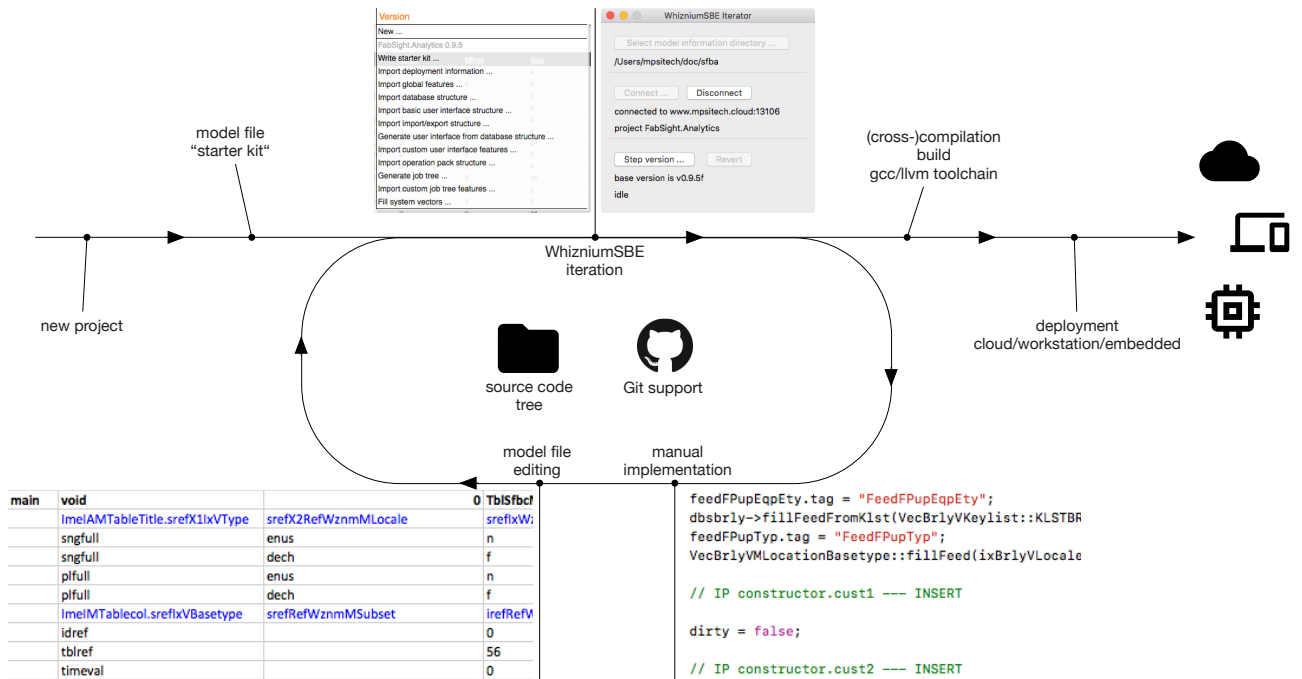
from May 2019



WhizniumSBE and WhizniumDBE scope



WhizniumSBE and WhizniumDBE development workflow



main	void		0	TblSfbcf	
	ImelAMTableTitle.srefX1lxVType	srefX2RefWznMLocale		srefXW:	
	sngfull	enus		n	
	sngfull	dech		f	
	plfull	enus		n	
	plfull	dech		f	
	ImelMTablecol.srefXVBasetype	srefRefWznMMSubset		ireRefV	
	idref			0	
	tblref			56	
	timeval			0	

```

feedFPupEqEty.tag = "FeedFPupEqEty";
dbsbrly->fillFeedFromK1st (VecBrlyVKKeylist::KLSTBR
feedFPupTyp.tag = "FeedFPupTyp";
VecBrlyVMLocationBasetype::fillFeed (ixBrlyVLocale
// IP constructor.cust1 --- INSERT

dirty = false;

// IP constructor.cust2 --- INSERT
    
```