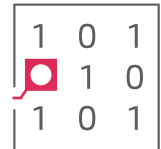


# WhizniumSBE und WhizniumDBE

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



MPSI

## Ihre Herausforderungen beim Umsetzen leistungsfähiger Embedded- und IIoT-Projekte

- Stetig wachsender Umfang der zu beherrschenden Themen: Multi-Threading, programmierbare Logik, Web-Technologien, Sicherheit, ...
- Aufwändige und teure Neu-Entwicklungen im Zuge der Digitalisierung
- Fachkräftemangel: Nachfrage nach qualifizierten Mitarbeitern ist größer als Angebot
- Wachsender Zeitdruck, „digitalisierte“ Produkte zur Marktreife zu führen
- Unnötige Folgekosten durch Schnellschüsse („technical debt“)

technische  
Details  
umseitig!



Unser revolutionärer Lösungsansatz:  
Automatische Quellcode-Generierung mit den WhizniumSBE/DBE  
Software-Entwicklungswerkzeugen



## WhizniumSBE (Service Builder's Edition)

Innovatives Werkzeug zur automatischen Quellcode-Generierung für kommunikative Embedded-Lösungen



- (Embedded) Linux, Windows, MacOS Ziele
- Multi-Threaded C++ Hauptprogramm
- Angehängte Datenbank (SQLite, MariaDB, ...)
- Vielsprachiges HTML5 User Interface
- API-Programm-bibliothek für einfaches Einbinden
- Standard IIoT-Protokolle: OPC UA, DDS
- Verteiltes Rechnen
- Einstellungen, Zugriffsverwaltung, Dateiarchiv, ...

Beispiel-Anwendungen: Anlagensteuerung, Sensorik, maschinelles Lernen, künstliche Intelligenz uvm.

## Ihre Vorteile



- Bis zu 70% schnellere Projektumsetzung senkt Entwicklungskosten und ermöglicht frühere Markteinführung
- Automatisches Coden von Nicht-Kern-Funktionen führt zu erheblicher Produktivitätssteigerung
- Saubere Quellcode-Struktur erleichtert Teamarbeit und Wartung
- Referenzfreier Quellcode: keine kostenpflichtigen oder versteckten Programm-bibliotheken werden benötigt
- Problemloses Mischen von automatisch generiertem Quellcode mit manuellen Ergänzungen
- Attraktives Software-as-a-Service Modell, Installationsoptionen „On-Premise“ und in der Cloud

## WhizniumDBE (Device Builder's Edition)

Innovatives Werkzeug zur automatischen Quellcode-Generierung für programmierbare Logik-Projekte



- FPGA- und Mikrocontroller-basierte Hardware-Ziele
- Befehlssatz umgesetzt als VHDL Zustandsautomaten
- Custom C++ Programm-bibliothek für den Host

Beispiel-Anwendungen: High-Speed Signalverarbeitung, Machine Vision uvm.

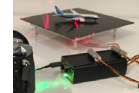
## Onboarding

- Zahlreiche Beispiel-Projekte
- Dokumentation und Demo-Code online
- Schulung und Consulting-Dienstleistungen
- Projektbearbeitung für Industriekunden

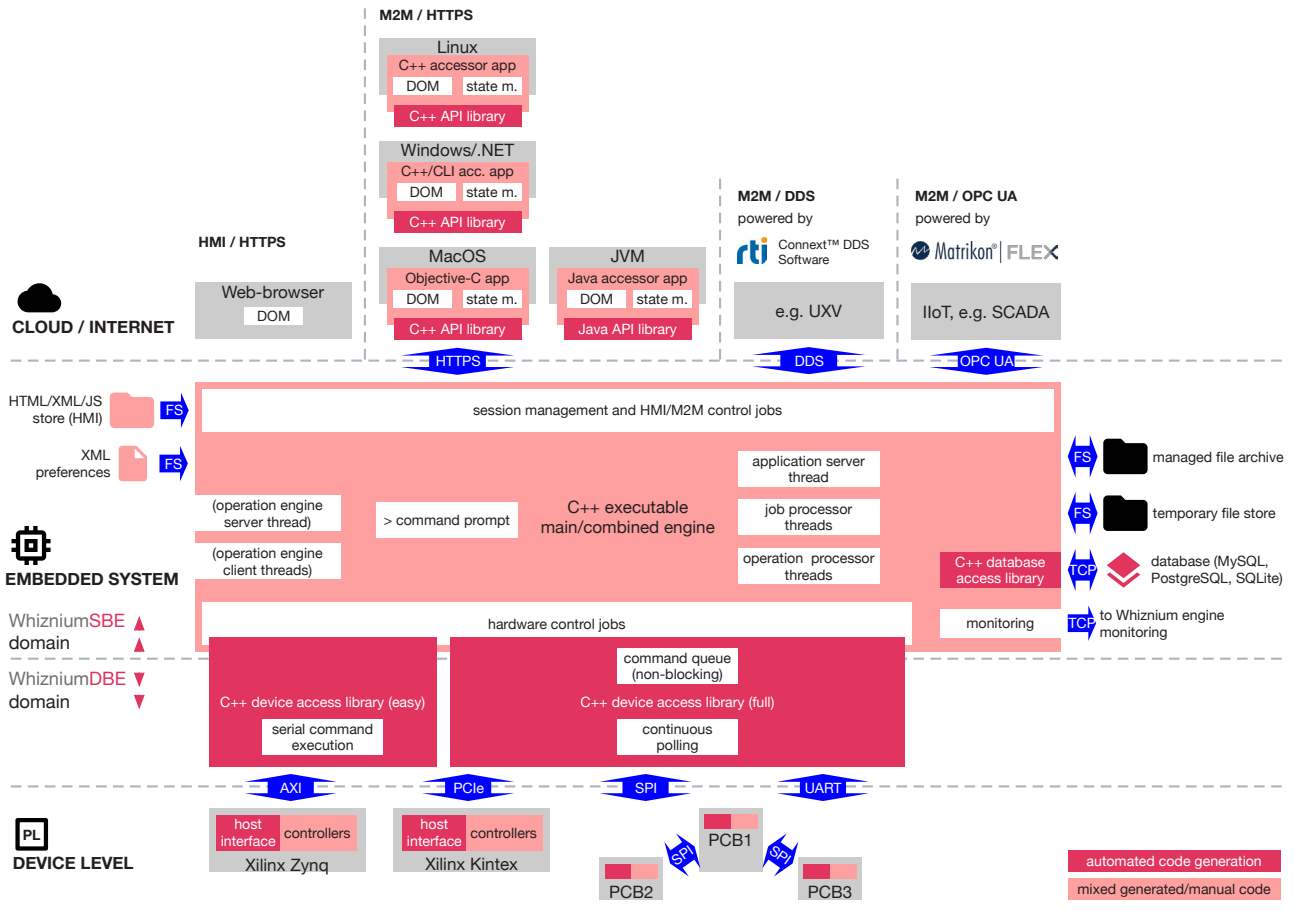
### Hardware Starter-Kit in drei Varianten Workstation | Embedded | FPGA

- Tabletop 3D Laserscanner
- Boards von Toradex / Digilent
- Testlizenz und WhizniumSBE/DBE Projekte

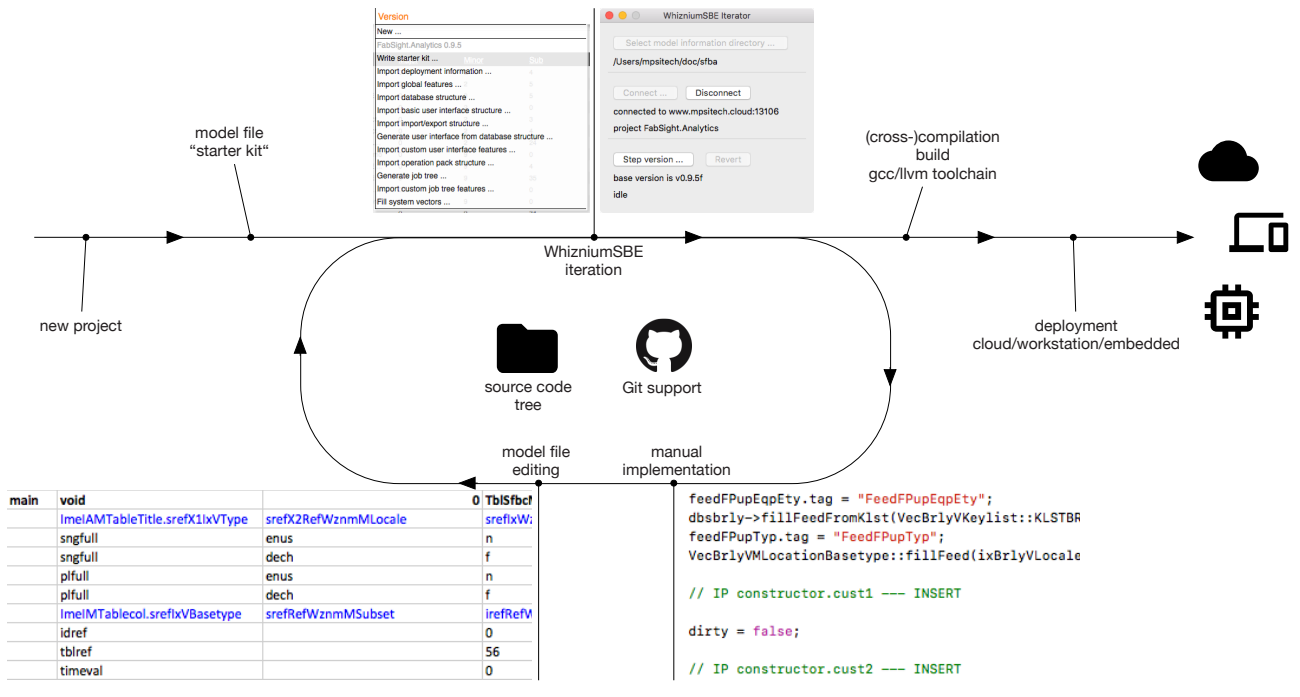
neu ab Mai 2019



# Funktionsumfang von WhizniumSBE und WhizniumDBE



# WhizniumSBE und WhizniumDBE im Entwickleralltag



main	void		0	TblSfbcf	
	ImelAMTableTitle.srefX1xVType	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
    
```