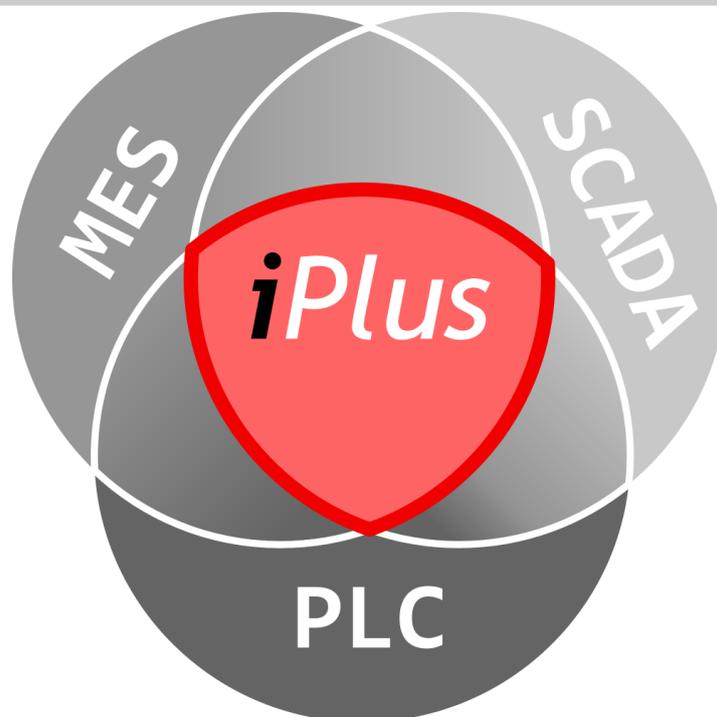


# *framework*



Die universelle Softwareplattform

# integrativ, produktiv...



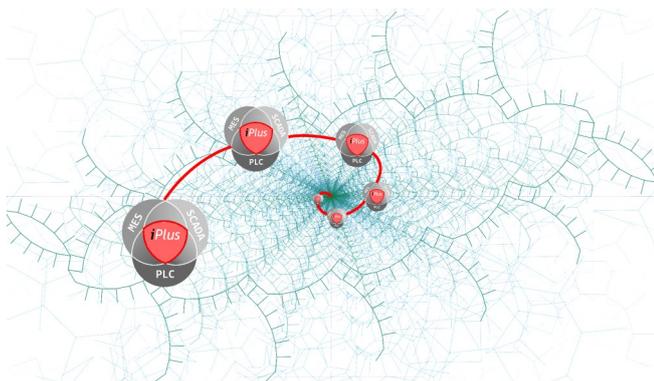
Das iPlus-Framework ist eine Softwareplattform, welche die datenbankbasierte mit der objektorientierten Anwendungsentwicklung und deren Präsentationsebene vereint. Es ist **zugleich** eine **Entwicklungs-** und eine **Laufzeitumgebung**.

Sie können damit z.B. transaktionsbasierte ERP- & MES-Systeme, mobile Lösungen, SCADA-Anwendungen, Automatisierungslösungen etc. mit einem einheitlichen Programmiermodell entwickeln. Ihre Lösungen können Sie aber auch als Softwarekomponenten bereitstellen, die wiederum andere Entwickler und Projektoren nutzen, um Ihren Endkunden eine integrierte Gesamtlösung zu liefern. Ein Projekteur muss dafür gar keine Programmierkenntnisse haben, weil er die Komponenten in der iPlus-Entwicklungsumgebung nur noch per drag & drop orchestrieren muss.

Dieser neue Ansatz ermöglicht zum einen eine enorme **Produktivitätssteigerung** und zum anderen eine noch nie dagewesene **Bedienerfreundlichkeit** und **Einfachheit** für Entwickler, Projektoren, Berater und Endanwender.

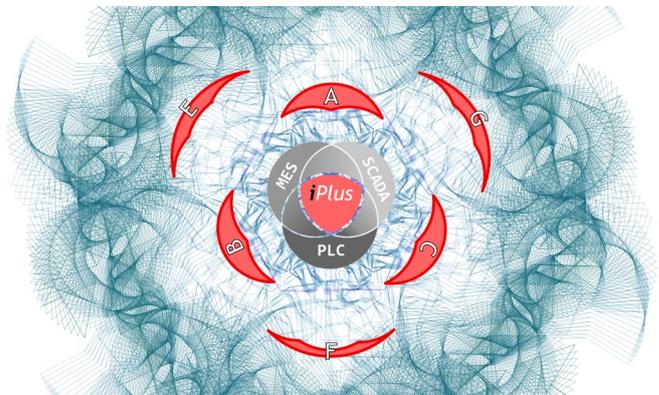
## .Architektur

Das Herzstück von iPlus ist ein objektorientiertes, komponentenbasiertes Framework, das auf dem .NET-Framework von Microsoft® aufsetzt. Das System ist vollständig **serviceorientiert** aufgebaut. Das bedeutet, dass alle Klassen bzw. Komponenten nur per Konfiguration in der Datenbank zum Leben erweckt werden. Alle Instanzen können **netzwerktransparent** miteinander kommunizieren. Auch die Programmierung der Komponenten erfolgt netzwerktransparent. Somit ergibt sich eine **grenzenlose Skalierbarkeit** des Systems und es können Multitier-Architekturen per Mausklick aufgebaut werden. Das System arbeitet zudem völlig **ereignisgesteuert**, um eine fast **echtzeitnahe** Prozessabarbeitung und Visualisierung zu ermöglichen.



## .Grenzenlose Erweiterbarkeit

Mit **Microsoft Visual Studio®** können Sie Ihre **eigenen Bibliotheken** entwickeln, die auf dem iPlus-Framework aufsetzen. Dabei müssen Sie sich nur an ein paar Regeln halten, um sogenannte Automation-Components erstellen zu können. Das Schöne dabei ist, dass Sie sich nur auf die Implementierung Ihrer Logik konzentrieren müssen und der Rest der Komplexität (Netzwerktransparentes Eventhandling, asynchrone Methodenaufrufe, Datenbankzugriff, Persistierung von Objektzuständen...) für Sie verborgen bleibt. Diese Abstraktion ermöglicht eine **universelle Programmierung** aller Klassen. Egal, ob Sie ein transaktionsbasiertes Businessobjekt (z.B. Lagerverwaltungskomponente), ein Automatisierungsobjekt (z.B. Ventil, Motor...), eine Kommunikationskomponente (z.B. ERP-Anbindung, Mailkomponente, Scanner...) oder etwas anderes programmieren wollen, das Programmiermodell ist immer dasselbe und **vor allem einfach!**



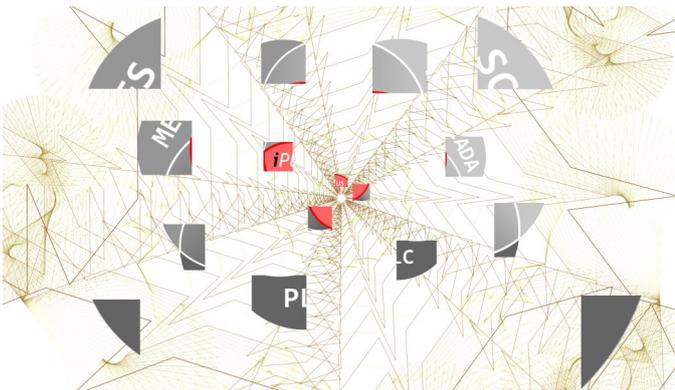
Für kleinere **projektspezifische Anpassungen** benötigen Sie kein Visual Studio, sondern entwickeln direkt in der **iPlus-Entwicklungsumgebung**. Die Entwicklungsumgebung besteht im Wesentlichen aus dem Projekt-Konfigurator, Oberflächendesigner, Workflowdesigner und C#-Editor. Sogar das Datenbankmodell (Entity-Framework) kann um Felder und komplexe, serialisierbare Klassen erweitert werden. Falls Ihnen das dennoch nicht ausreicht, können Sie natürlich auch Ihr eigenes Datenbankmodell hinzufügen. Dadurch, dass die iPlus-Entwicklungsumgebung mit dem iPlus-Framework programmiert ist, können Sie diese ebenfalls objektorientiert **erweitern**, und sich selbst Werkzeuge bauen, die ihr Tagesgeschäft beschleunigen und die Projektierungszeit wesentlich verkürzen.

# intelligent, innovativ...



## .Ihre eigene Branchenlösung

Entwickeln Sie Ihre eigene Branchenlösung, die Sie entweder für Ihre eigenen Projekte verwenden oder verkaufen Sie Ihre iPlus-Bibliotheken an andere Firmen. iPlus hat einen **integrierten Lizenzmanager**, der Ihnen das erleichtert.



## .Update- und Upgrade

Mit dem **integrierten Paketverwaltungssystem**, können Sie problemlos Ihre Bibliotheken, Projektkonfiguration, Designs, etc. als Updatepaket exportieren. Somit können Sie entweder Ihren Lizenznehmern Updates zur Verfügung stellen oder Ihren Endkunden Ihre projektspezifischen Änderungen ausliefern, die Sie zuvor auf Ihrem Entwicklungssystem getestet haben. Sogar direkte Update-Transporte von Datenbank zu Datenbank sind möglich.

Aufgrund der besonderen Architektur des iPlus-Frameworks und der losen und dynamischen Kopplung der Bibliotheken, können iPlus-Updates, die wir ausliefern, unabhängig von Ihren Updates eingespielt werden und das System läuft anschließend problemlos wieder hoch.

## .Integration und Kooperation

Aufgrund der **serviceorientierten** Architektur, können verschiedene Hersteller Ihre eigenen Lösungen entwickeln, die später gemeinsam auf einem Zielsystem laufen, **ohne** dass **eine Kompilierung zu einem Gesamtsystem** notwendig ist. Es ist vergleichbar mit einem herkömmlichen Paketverwaltungssystem wie z.B. nuget jedoch mit den wesentlichen Unterschieden, dass

- die Softwarekomponenten nicht durch eine Endkompilierung zusammen verbunden werden müssen,

- dass ein Softwarepaket auch nur deklarativ sein kann (keine .NET-Assemblies, sondern nur Projektkonfigurationsdateien und GUI-Beschreibungen XAML),
- die Pakete während der Laufzeit (z.B. in einem Produktivsystem) hinzugefügt werden können.

### # Daraus ergeben sich folgende Vorteile:

Es können **unterschiedliche Parteien** (Projektueure, Berater, Entwickler, Teams) unabhängig voneinander an einer Gesamtlösung arbeiten.

Die **Erweiterung** der Gesamtlösung kann für zeitkritische Systeme zum Teil sogar **während des Produktivbetriebs** stattfinden. Wenn dann ein geeignetes Zeitfenster zur Verfügung steht, können die iPlus-Dienste bereits mit der fertig konfigurierten Einstellung neu gestartet werden.

Integriertes Change-Request-Management: Projektierungen die auf Testsystemen erstellt wurden (lokaler Rechner, gemeinsames Testsystem...), können per Export-/Import auf das Produktivsystem übertragen werden.

## .GUI & SCADA

# Die **vektorbasierte 2D-/3D-Oberfläche** (WPF) wird dynamisch aus der Datenbank generiert (XAML). Designs können entsprechend der Objektorientierung überschrieben werden. Die gesamte Oberfläche (Masken, Steuerelemente...) kann während der Laufzeit **individuell angepasst** werden und ist trotzdem updatefähig. Außerdem ist sie adaptiv und passt sich jeder Bildschirmauflösung an.

# **Hierarchisches** und einheitliches **Alarmsystem** für alle Komponenten. Unbegrenzte Alarmhistorie und Verknüpfbarkeit mit Anwendungsdaten (Produktionsauftrag, Batch, Materialwirtschaft..)

# **Pozessvariablenlogging**. Jede Eigenschaft einer beliebigen Instanz kann ereignisbasiert geloggt und in einem **Kurvendiagramm** dargestellt werden. Das Kurvendiagramm kann sowohl Live- als auch Historienwerte auf verschiedenen Skalen darstellen. Das Steuerelement zur Darstellung der Kurven bietet sehr viele Einstell- und Präsentationsmöglichkeiten.

# einfach, einheitlich...



## .Workflows

# **Grafischer Workflow-Designer** zur Programmierung aller Arten von Prozessabläufen. Damit können Sie Ihre eigenen Geschäftsprozesse jeglicher Art abbilden. (z.B. Produktionsprozesse, Logistische Prozesse, Maschinensteuerungen...)

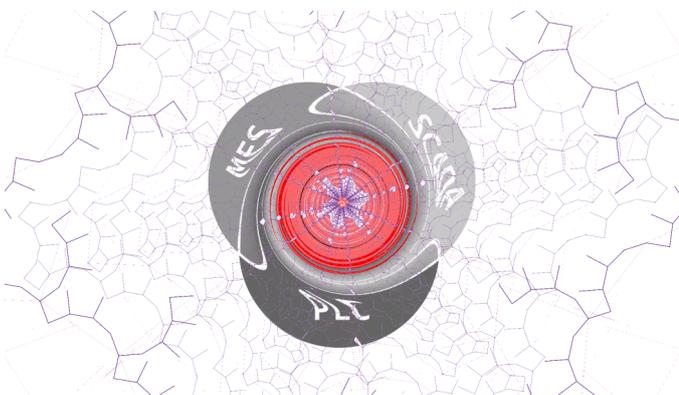
# **Grafische Programmablauf-Historie**. Sämtliche Prozessschritte, die bei der Abarbeitung eines Workflows ausgeführt werden, werden samt Ihren Parametern protokolliert. Mit einem Log-Viewer können Sie anschließend das Protokoll in Gantt-Darstellung anschauen. Ein komfortables und umfangreiches Suchsystem hilft dem Anwender Prozessprobleme zu erkennen und zu optimieren.

# **Umfangreiche Diagnose- und Logmöglichkeiten**. Objekt-Explorer (Live-Debugger), der den Objektzustand aller Instanzen samt ihren Eigenschaften (ob Server- oder clientseitig) visualisieren und manipulieren kann.

## .PLC

# **OPC-UA, Modbus-TCP und RFC1006(Simatic®)-Treiber**.

# **Vollständige Beckhoff®-TwinCAT3®-Integration**. Wir haben unser iPlus-Automation-Component-Modell auch in TwinCAT3® objektorientiert umgesetzt. Somit können Sie Ihr iPlus-Projekt fast 1:1 in TwinCAT® abbilden. Unser TwinCAT-Projekt-Synchronisator hilft Ihnen dabei und generiert bzw. synchronisiert Ihr gesamtes iPlus-Projekt mit dem TwinCAT®-Projekt. Letztendlich ersparen Sie sich 80% der SPS-Programmierzzeit und 100% der Konfigurationszeit.



## .Sonstige Funktionen

# **XAML-basierter grafischer Reportgenerator**. Es können nicht nur Reports aus Datenbankdaten erstellt werden, sondern aus dem gesamten verfügbaren Objektmodell (Livewerte von Komponenten, serialisierbare Objekte,...)

# **Mehrsprachigkeit**. Es können beliebig viele Sprachen hinzugefügt werden. Es werden nicht nur die Funktionen übersetzt, sondern es können auch Felder aus Anwendungstabellen übersetzt werden.

# **Gruppenbasiertes Rechtssystem**. Es können Schreib- und Sichtbarkeitsrechte für alle Objekte, Eigenschaften und Methoden vergeben werden.

# **Rollenbasiertes Online-Dokumentationssystem**. Die Dokumentation kann aus unterschiedlichen Rollen heraus dargestellt und verknüpft werden.

# **Corporate Design**. Sie können den Stil und das Design von iPlus nach Ihren Vorstellungen und dem CI Ihrer Firma anpassen.

Markenhinweise:

Microsoft®, .NET, SQL-Server® und Visual Studio® sind eingetragene Marken der Microsoft Corporation.

Siemens® und Simatic® sind eingetragene Marken der Siemens AG.

Beckhoff® und TwinCAT® sind eingetragene Marken der Beckhoff GmbH & Co. KG.

## Sonstige Informationen

Download und Dokumentation auf

<http://www.iplus-framework.com>



Gipsoft d.o.o.  
Repovec 25B  
HR-49210 Zabok

E-Mail: [info@iplus-framework.com](mailto:info@iplus-framework.com)

**gipsoft**