Mechatronic
=medical solutions

Automatisiertes Testen von Embedded Systems in der Medizingeräteentwicklung

MedConf 2011 – München
06.10.2011

Mechatronic AG
Thomas Jetter & Sven Rippel

Mechatronic AG

Mechatronic
=medical solutions

- Gegründet 1987
- 70 Mitarbeiter
- Branche: Medizintechnik
- Unser Leistungsspektrum:
  - Systementwicklung
  - Hardwareentwicklung
  - Softwareentwicklung
  - Production Engineering
  - Fertigung

Höhn
(Niederlassung)

Darmstadt
(Hauptsitz)

Automatisiertes Testen von Embedded Systems in der Medizingeräteentwicklung

Agenda

1. Einleitung
2. Vorüberlegungen
3. Konzeption
4. Realisierung
5. Praktisches Beispiel
6. Verifizierung/Validierung
7. Erkenntnisse und Ausblick

Warum automatisiert testen?

- Eindeutigkeit
- Reproduzierbarkeit
- Bessere Testabdeckung
- Zeitersparnis bei Wiederholung
- Effizientere Entwicklung
- Stresstests

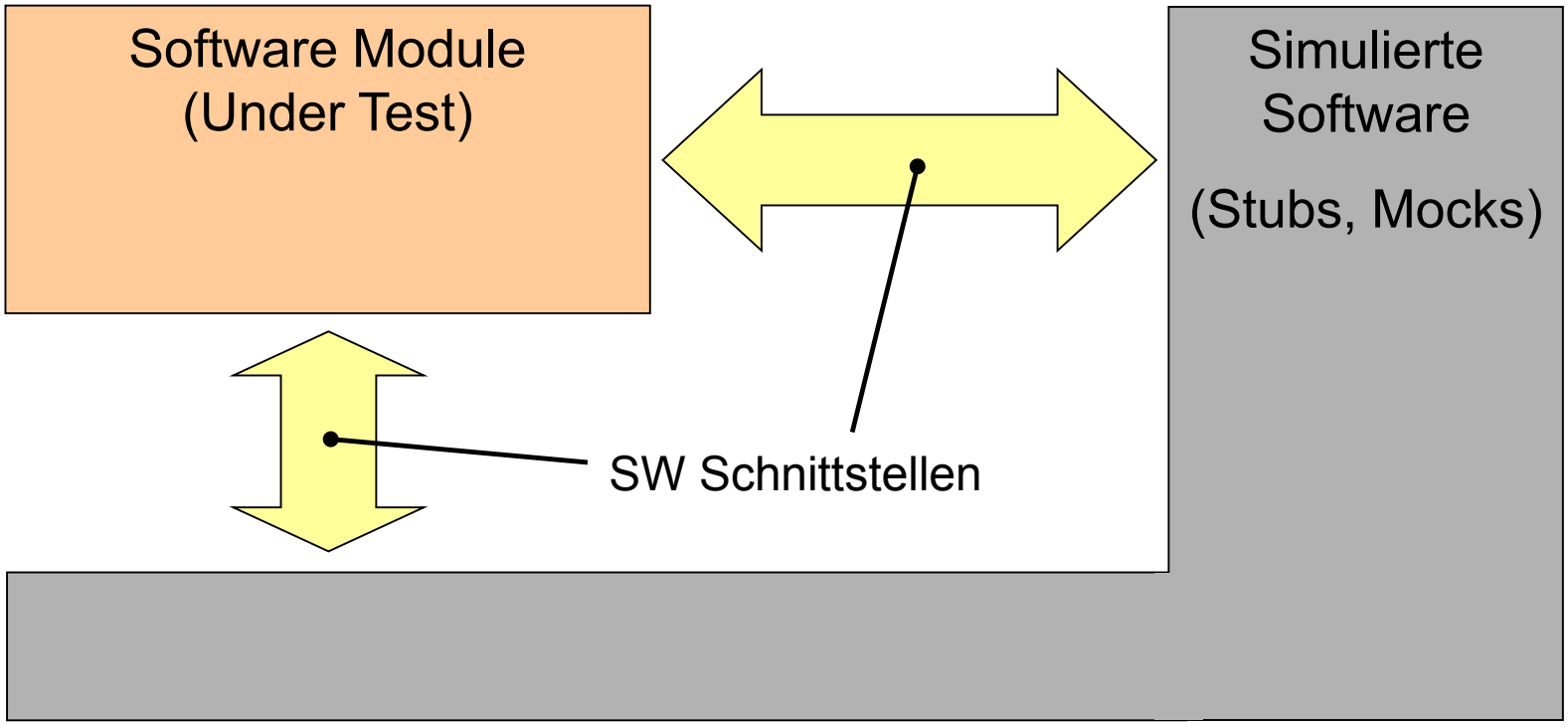
Begriffsklärungen

Software-Teststufen

- **Unit-Tests**
  Einzelne, losgelöste Software-Module
- **Integration Tests**
  Von einander abhängige Software-Module als Einheit
- **System Tests**
  Integriertes Software-System auf Ziel-Hardware

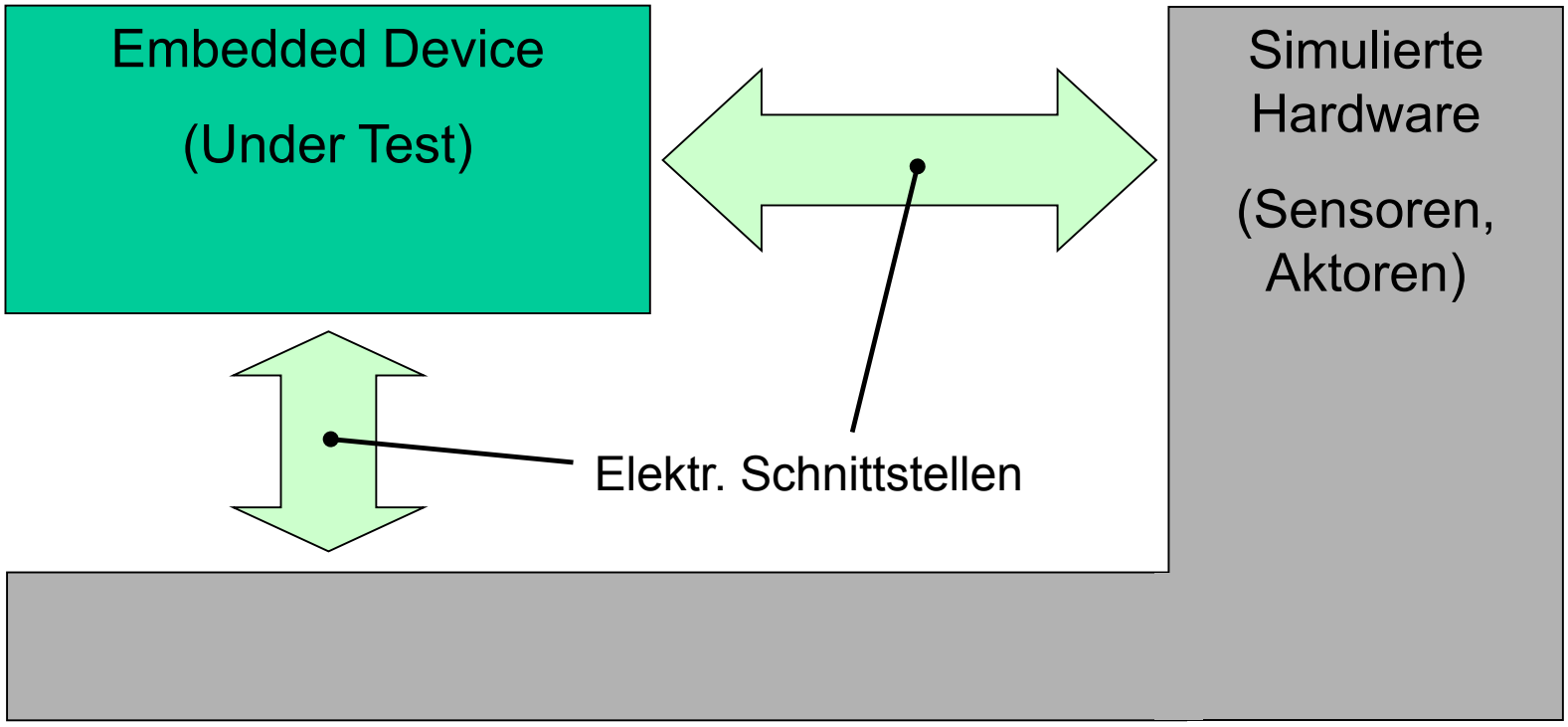
Begriffsklärungen

SiL (Software in the Loop)



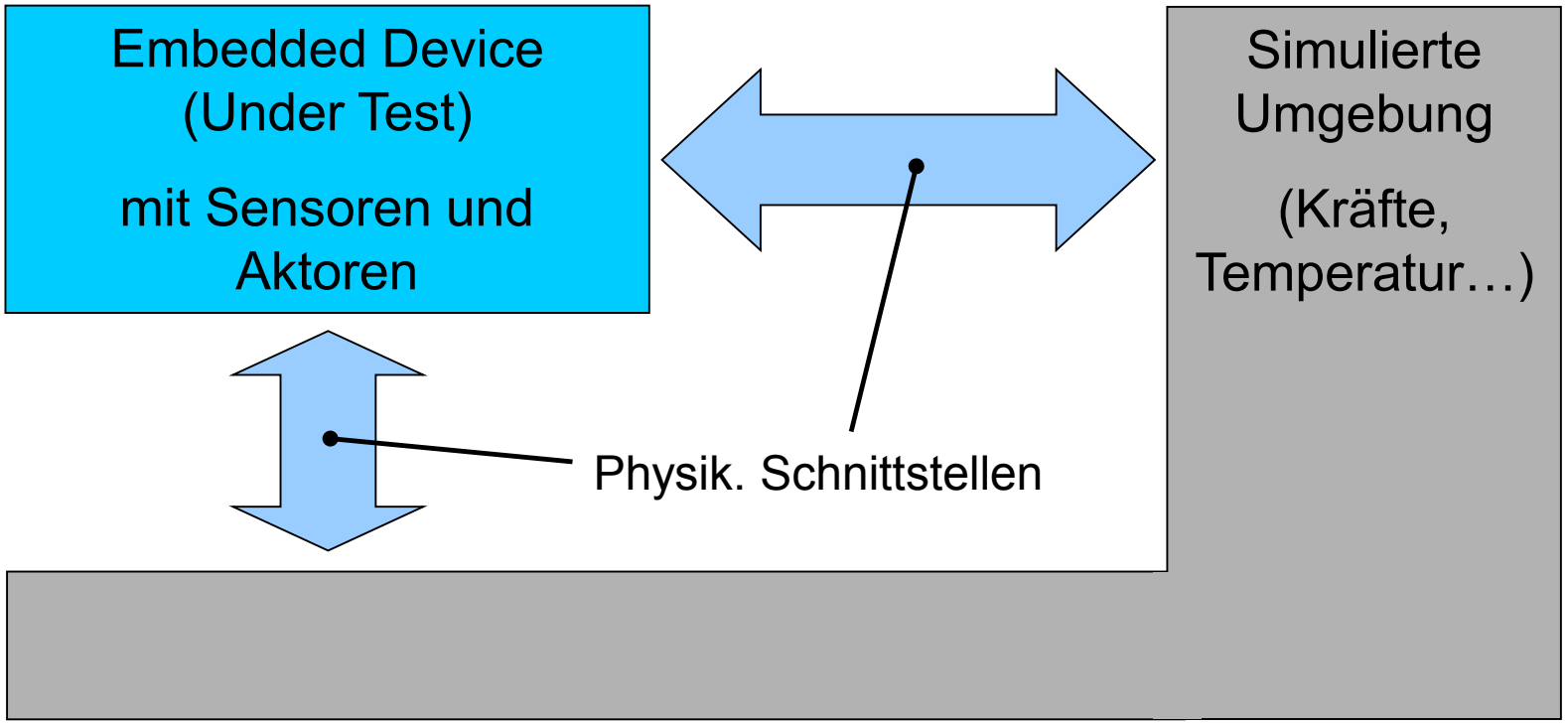
Begriffsklärungen

HiL (Hardware in the Loop)



Begriffsklärungen

EiL (Environment in the Loop)



Automatisiertes Testen von Embedded Systems in der Medizingeräteentwicklung

Agenda

1. Einleitung
2. Vorüberlegungen
3. Konzeption
4. Realisierung
5. Praktisches Beispiel
6. Verifizierung/Validierung
7. Erkenntnisse und Ausblick

Kernanforderungen

- Systemtests auf Zielplattform (SiL, HiL, EiL)
- Projektübergreifender Automatisierungsansatz
- Gute Integration externer Hardware
- Einfache, transparente Skripte
- Flexible Dokumentengenerierung
- An interne Prozesse anpassbar

Kernanforderungen

Warum auf dem Target testen?

- Zeitkritische Abläufe
- Interrupt-Handling
- Compilerfehler
- Speicherüberläufe
- Integration mit Hardware

Abgrenzung

- Kein Unit-Level Test Tool
- Kein Tool für Produktionstests

Marktanalyse

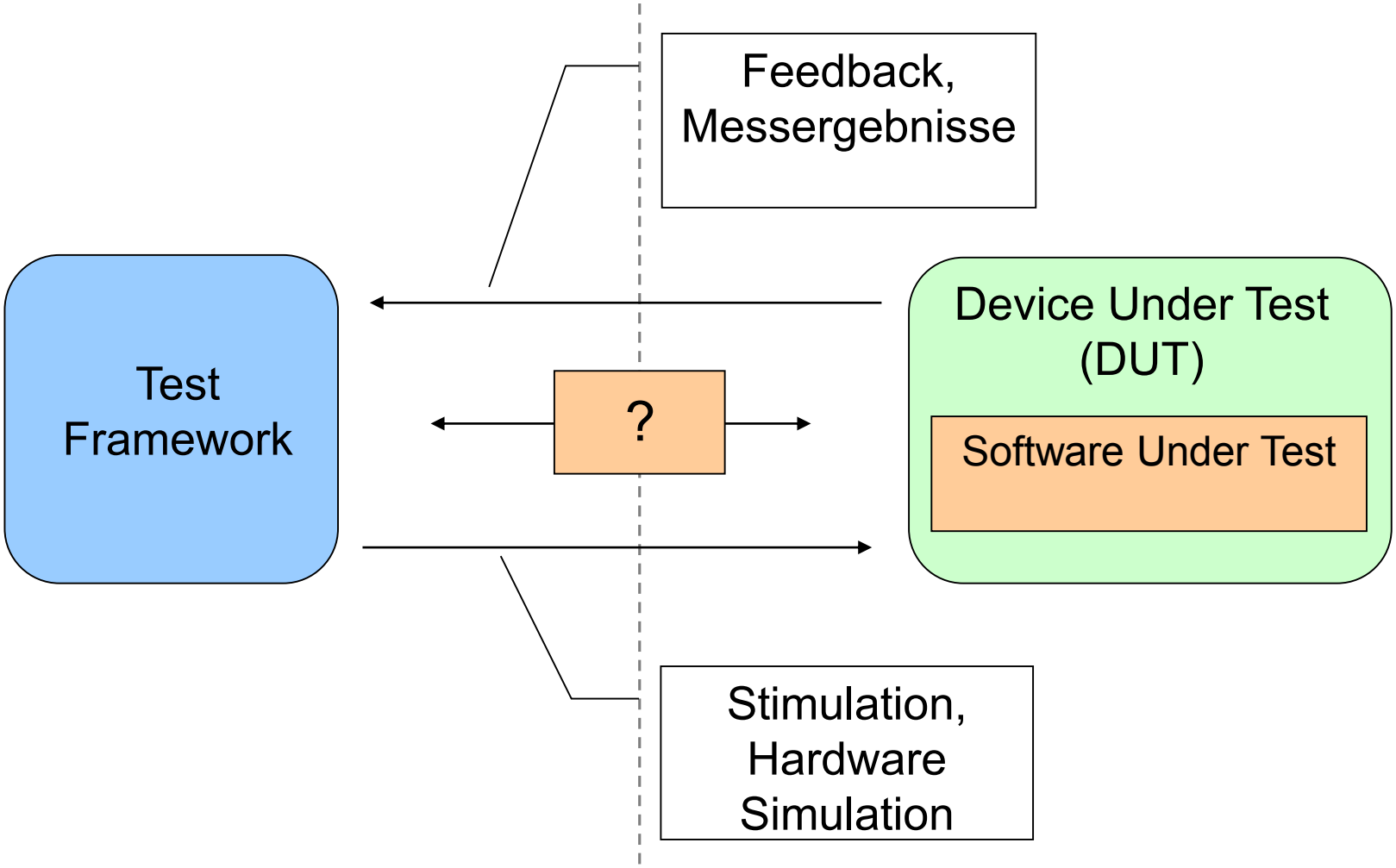
- Viele Tools mit SiL Testansatz
  - Relativ einfach zu realisieren
  - Sehr universell einsetzbar
  - Oftmals stark auf Unit-Level Tests fokussiert
- HiL/EiL Testansatz wenig vertreten
  - Anforderungen sind zu unterschiedlich
  - Externe Hardware ist ein Muss
- **Fazit: Eigenentwicklung**

Zusätzliche Herausforderungen bei Eigenentw.

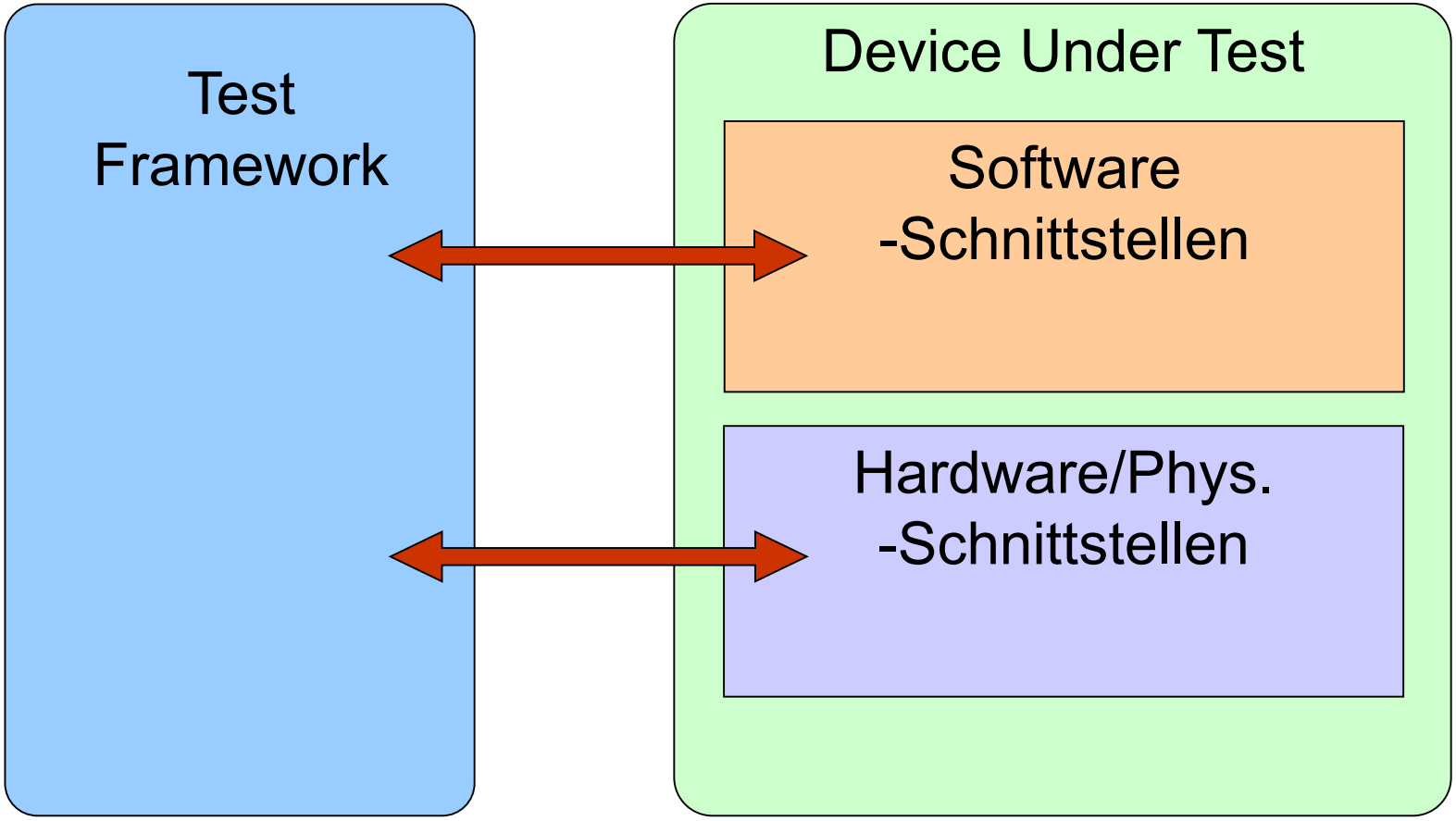
- Budgetierung (eigenes Entwicklungsprojekt)
- Entwicklungsdauer (verzögerte Verfügbarkeit)
- Tragfähiges Systemkonzept

Automatisiertes Testen von Embedded Systems in der Medizingeräteentwicklung

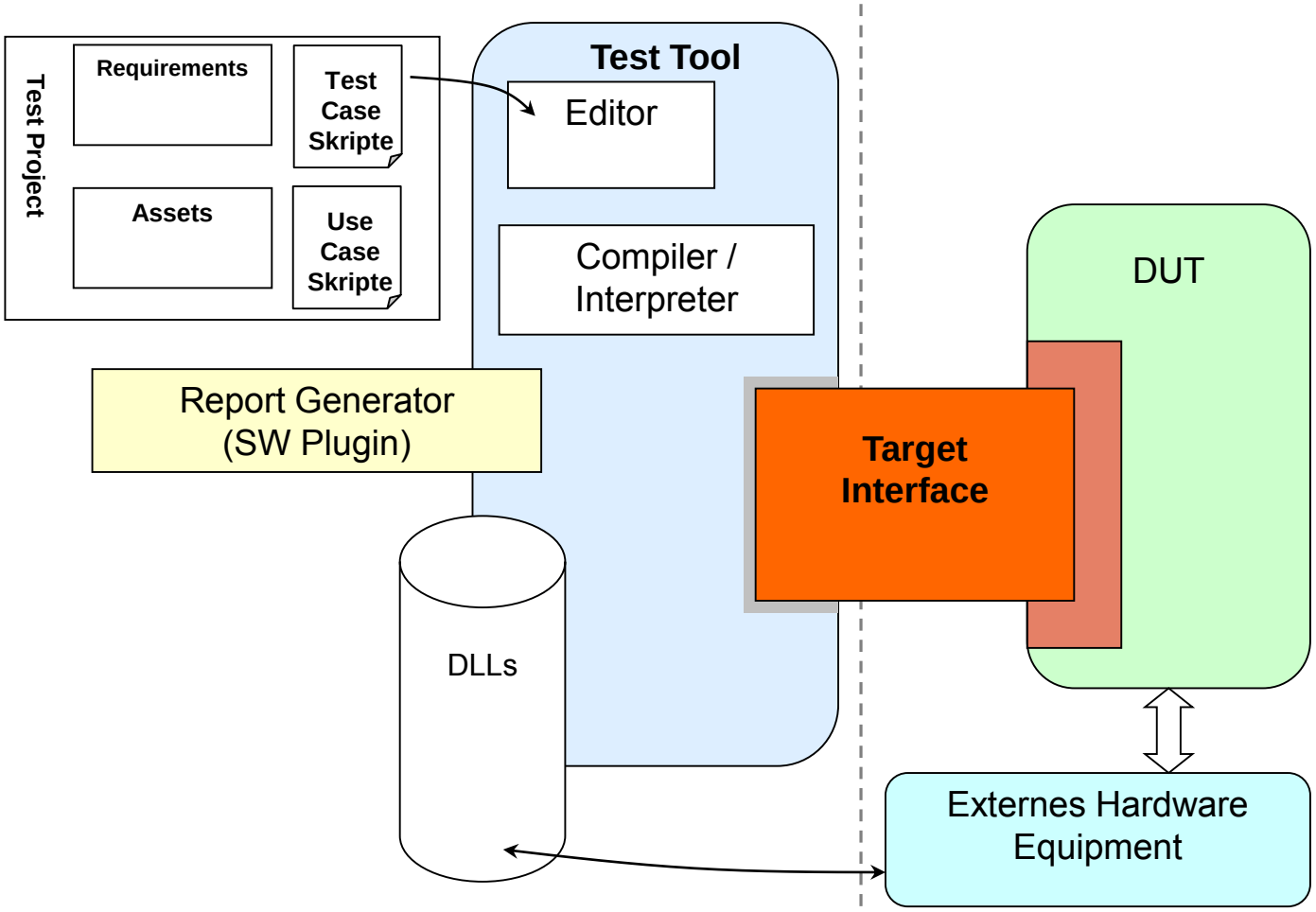
Grundkonzept



Lösungsansatz



Detailkonzept



Automatisiertes Testen von Embedded Systems in der Medizingeräteentwicklung

Skripte

Anforderungen an die Skripte

- Einfach nachvollziehbar und zu reviewen
- Direkt im Test-Tool editierbar,
  kein externer Editor (Usability)
- Zugriff auf Referenzdaten (Assets)
- Hinzufügen von Attachments zum Report zur Ausführungszeit
- Obligatorische Pass/Fail Entscheidung
- Textuelle Testbeschreibung (für Protokoll und Report)

Skripte

Vereinfachung der Skripte

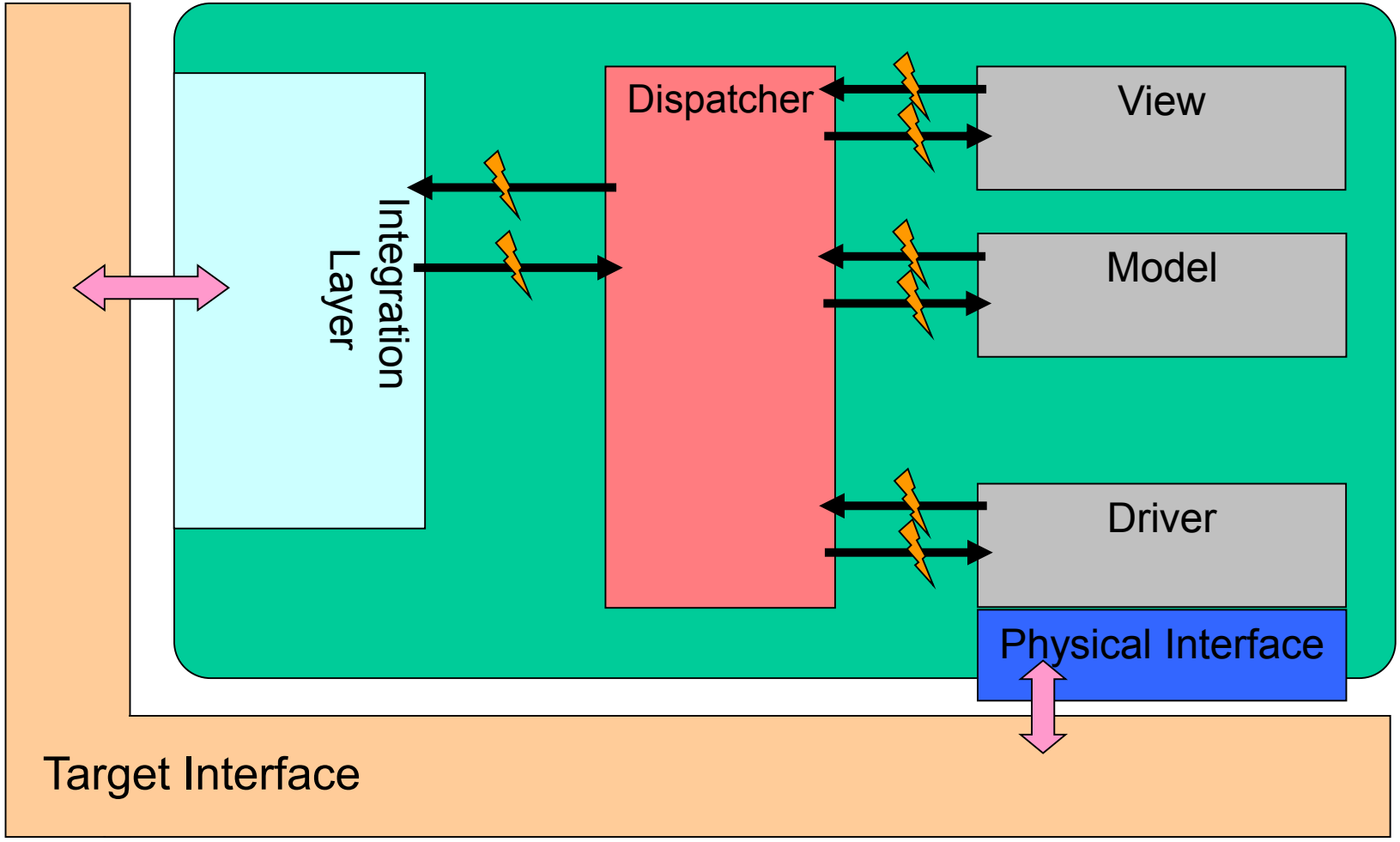
- Soviel Komplexität wie möglich auslagern:
  - Target Interface
  - Bibliotheken
  - Use Case Skripte
- Verzicht auf schwer zu durchschauende oder schwer zu überblickende Sprachkonstrukte (z.B. Lambda Expressions, Linq, Delegaten, Events)

Skripte

```
bool Run()
{
    var currents_uA = new double[] {12.5, 14.8};

    foreach(var referenceCurrent in currents_uA)
    {
        USECASE.ApplyCurrent(referenceCurrent);

        var measuredCurrent = TARGET.MeasureCurrent();

        ATTACH(referenceCurrent, "ReferenceCurrent [µA]");
        ATTACH(measuredCurrent, "MeasuredCurrent [µA]");

        if (measuredCurrent != referenceCurrent)
        {
            return FAIL;
        }
    }

    return PASS;
}
```

Toleranzgrenzen durch Operatorüberladung

Schnittstelle DUT ↔ Target Interface



Testdurchführung

- Testausführung aus dem Tool heraus
- Auswahl an Testcases festlegbar
- Speicherung der Informationen in einer Datei
- Reportgenerierung auf Basis dieser Daten

Anforderungen Report-Generator

- Verschiedene Formate, z.B. WORD und HTML
- Verwendung von Vorlagen
- Protokoll- und Reportgenerierung zur Freigabe
- Trace-Matrizen
- Test Summary

Anforderungen Report-Generator

- Test Case Einzelansicht
  - Pass/Fail Information
  - Textuelle Beschreibung
  - Traces zu den Requirements
  - Visualisierung der Attachments

Automatisiertes Testen von Embedded Systems in der Medizingeräteentwicklung

Das Produkt

- Mobiles In-Vitro Diagnosegerät
- Teil einer Gerätefamilie
- Ca. 1.000 Stück p.a.
- 32 Bit µC
- Neue Messtechnologie

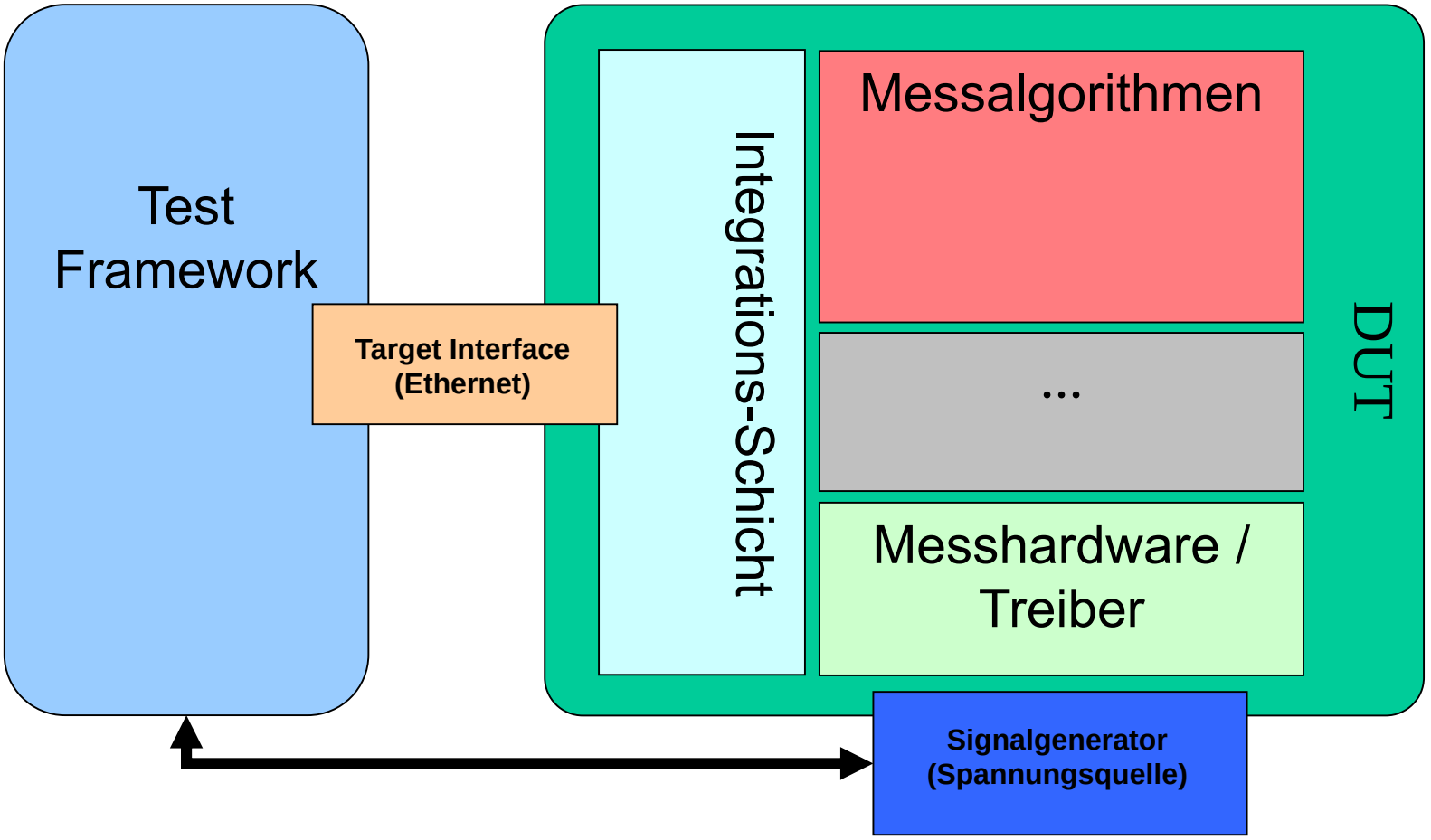
Rahmenbedingungen

- Kommunikationsschnittstelle und Protokoll durch Kunden vorgegeben (Gerätefamilie)
- Software auf bestehender Codebasis aufbauend
- Viele manuelle Testfälle für Gerätefamilie verfügbar und wieder verwendbar
- Beschränkung der Tests auf die neuen Messalgorithmen
- Implementierung der Messalgorithmen für automatisiertes Testen im finalen SW System optimiert

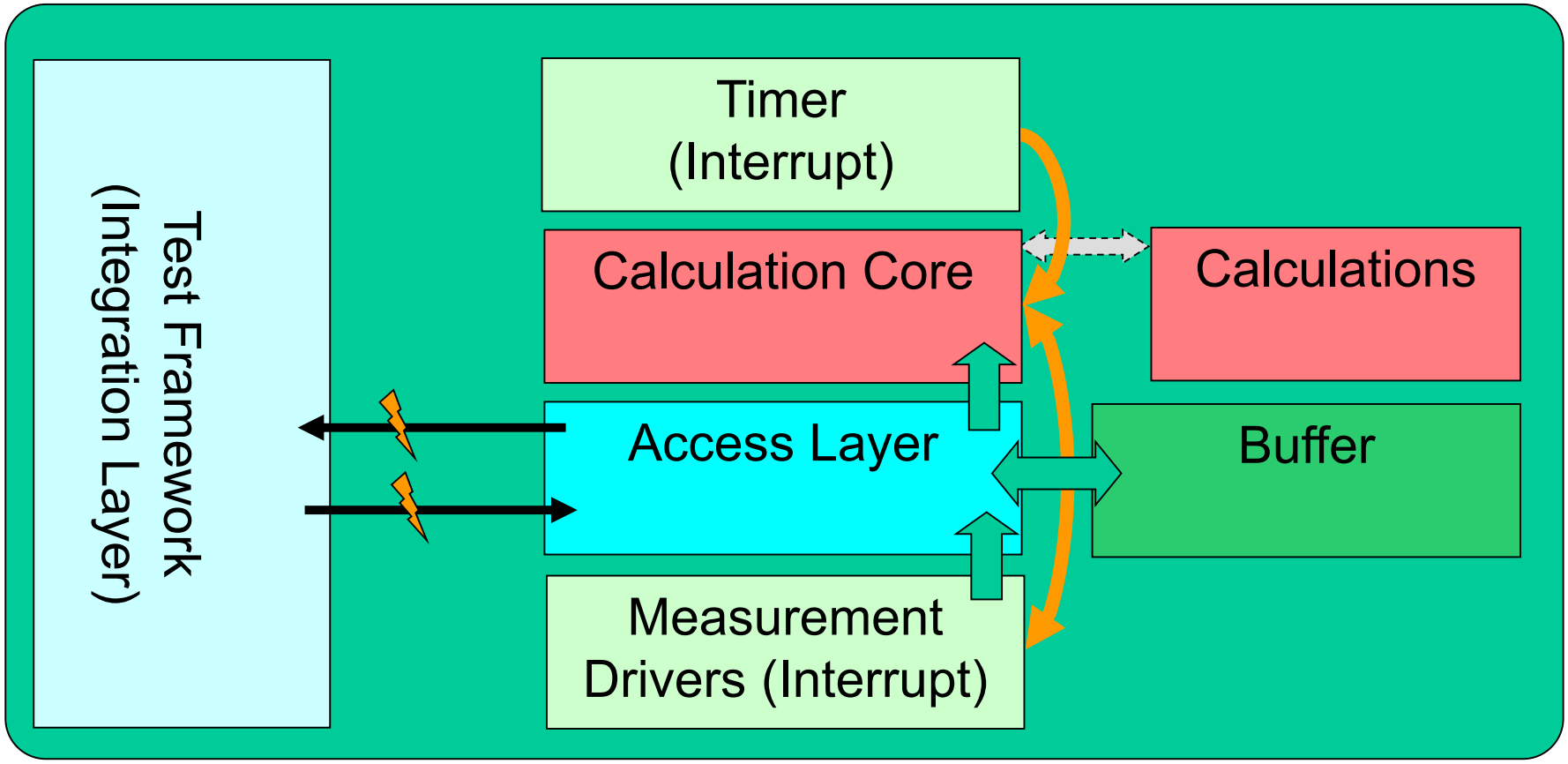
Besonderheiten der zu testenden Algorithmen

- Messwertreihen bestehend aus mehreren hundert Datenpunkten
- Komplexe Berechnungen in Floating Point Arithmetik
- Zeitkritisch, Interrupt getrieben
- Umfangreiche Mechanismen zur Fehlererkennung
- Temperaturkompensation

Integration in das Framework



SW Architektur



Kernfunktionen des Target Interfaces

- Schreiben und Lesen von Messwertreihen
- Einspeisung simulierter Temperaturwerte
- Anstoßen von Messungen
- Diversitäre Implementierung der Berechnungen in 128 Bit Gleitkommaarithmetik
- Komplexe Vergleichsoperatoren
- Manipulation von Messwertreihen

Erfahrungen

- Permutationen: ca. 7000 in 12 Stunden
- Stabilitätsnachweis frühzeitig und wiederholt erbracht
- Skripterstellung < 1 Woche
- Erstellung Target Interface ca. 1 Monat
- Minimaler Aufwand für weitere Produkte der Familie

Automatisiertes Testen von Embedded Systems in der Medizingeräteentwicklung

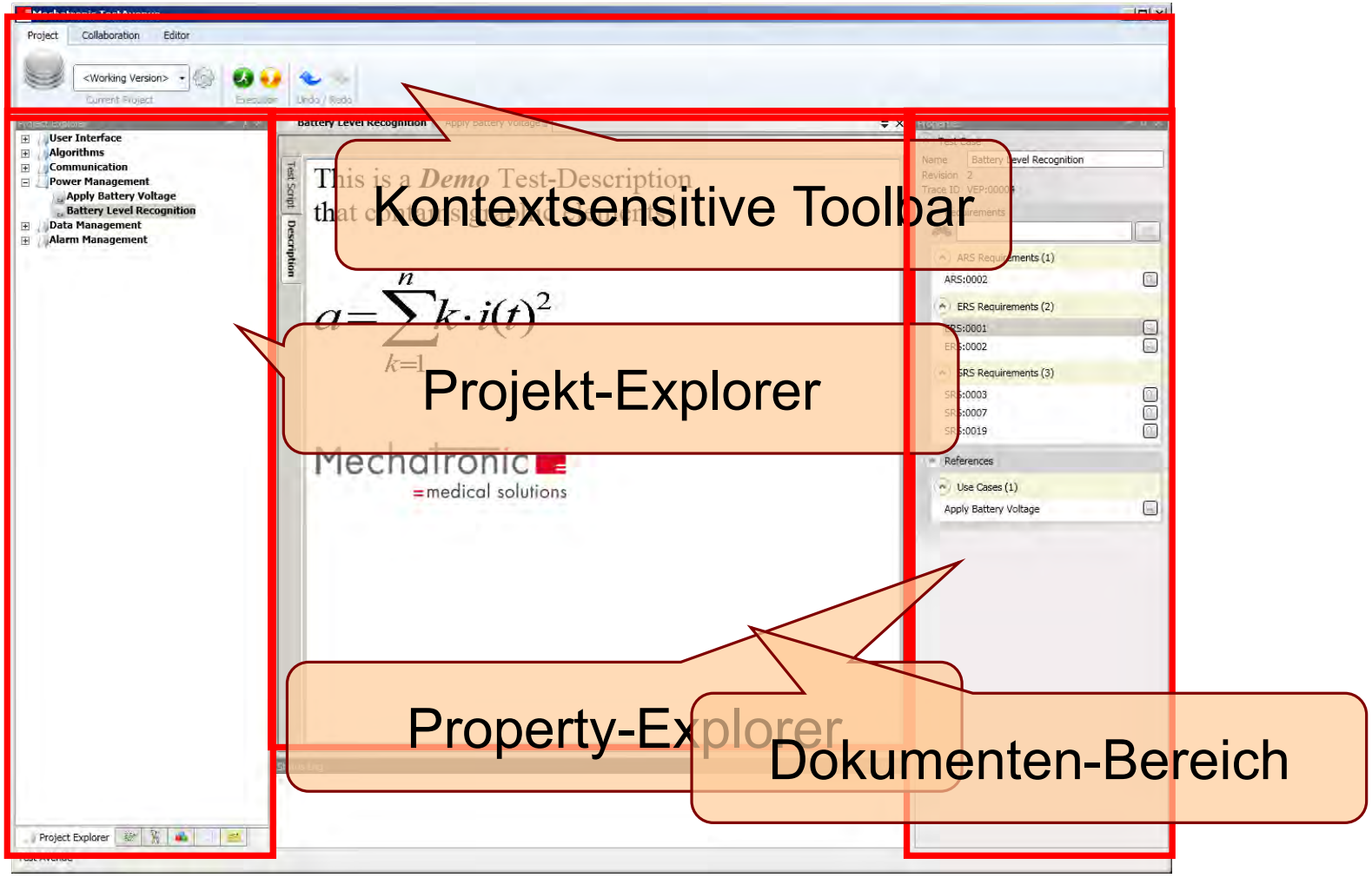
Tool Allgemein

- Fokus auf Kernfunktionen
  - Editor
  - Compiler
  - Test Execution
  - Report Provider
  - Tracing

Projektkontext

- Target Interface
  - Test-Skript
  - Ggf. manueller Testplan
  - Review
- Skripte und Assets
  - Review
- Libraries
  - Test-Skript
  - Unit-Test und Review

Screenshots



Automatisiertes Testen von Embedded Systems in der Medizingeräteentwicklung

Erkenntnisse & Ausblick

- Erweiterungen Test-Tool:
  - Beschreibungen als RTF
  - Multi-User Fähigkeit
  - Integriertes Konfigurationsmanagement
  - Plug-In Ansatz erweitern
  - Verbesserung der Usability
- Code Basis (Embedded)
  - Generische Eventschnittstellen

Resümee

- Erwartete Vorteile bestätigt
- Flexibles Mischen von SiL, HiL und EiL möglich
- Testansatz vollständig skalierbar
- Tool projektübergreifend etabliert

Richtige Entscheidung!


Mechatronic
=medical solutions