

# C-HIL-Integration für die Entwicklung von Niederspannungsnetz-Regelungssystemen

Mario Faschang  
ICT – TU Wien

Alfred Einfalt, Ralf Mosshammer  
Siemens AG Österreich

Roman Schwalbe  
AIT – Energy Department

## Regelalgorithmen-Entwicklungsprozess

Das Forschungsprojekt DG DemoNet – Smart LV Grid verfolgt die Maximierung des Anteils von erneuerbarer Energie im Niederspannungsnetz auf Basis gesteuerter Netzkomponenten wie PV-Wechselrichter, Elektrofahrzeug-Ladestationen und unter Last schaltbarer Transformator-Stufensteller.

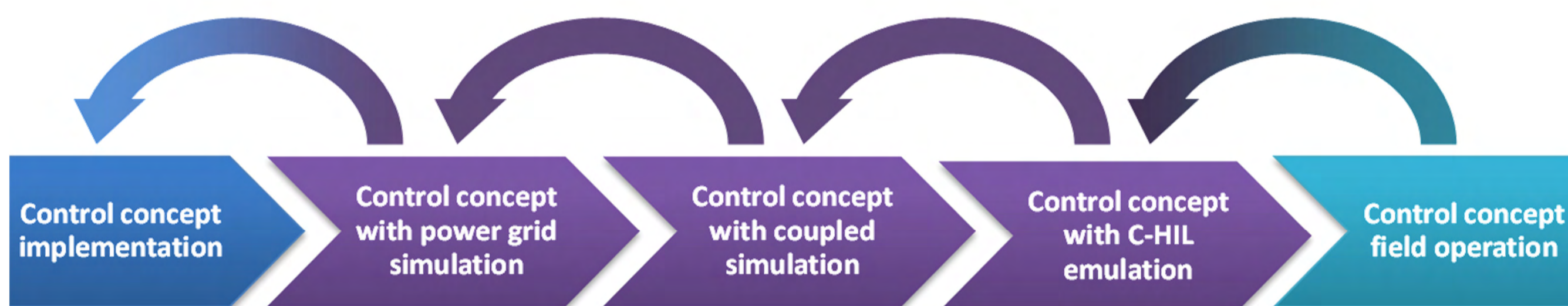


Abb. 1: Agiler Prozess für die Entwicklung der Regelalgorithmen

Zur Steuerung dieser aktiven Netzkomponenten wurden Regelalgorithmen mithilfe des in Abb. 1 dargestellten, mehrstufigen agilen Entwicklungsprozesses entwickelt. Der Prozess führt zu einer stetigen Risikoreduktion und begleitet den Algorithmus vom ersten Entwurf bis zum aktiven Feldebtrieb.

## Controller-Hardware-in-the-Loop Betrieb

Ein wichtiger Punkt für die Evaluierung der Regelansätze ist der vierte und somit letzte Schritt von der Inbetriebnahme im Feld: **Der Controller-Hardware-in-the-Loop (C-HIL) Betrieb.** Dabei wird der Regelalgorithmus bereits auf der Feld/Ziel-Hardware betrieben, arbeitet zu Testzwecken jedoch noch in einer simulierten Umgebung. Das Ziel ist schlussendlich der Betrieb des getesteten Reglers auf dem Zielsystem (Industrie PC mit UNIX-Distribution) im Feld (vgl. Abb.2).

Für diese Implementierung des C-HIL Betriebs gibt es im Projekt unterschiedliche Ausbaustufen, deren Basis das Co-Simulationssetup (vgl. Abb. 3) bildet.

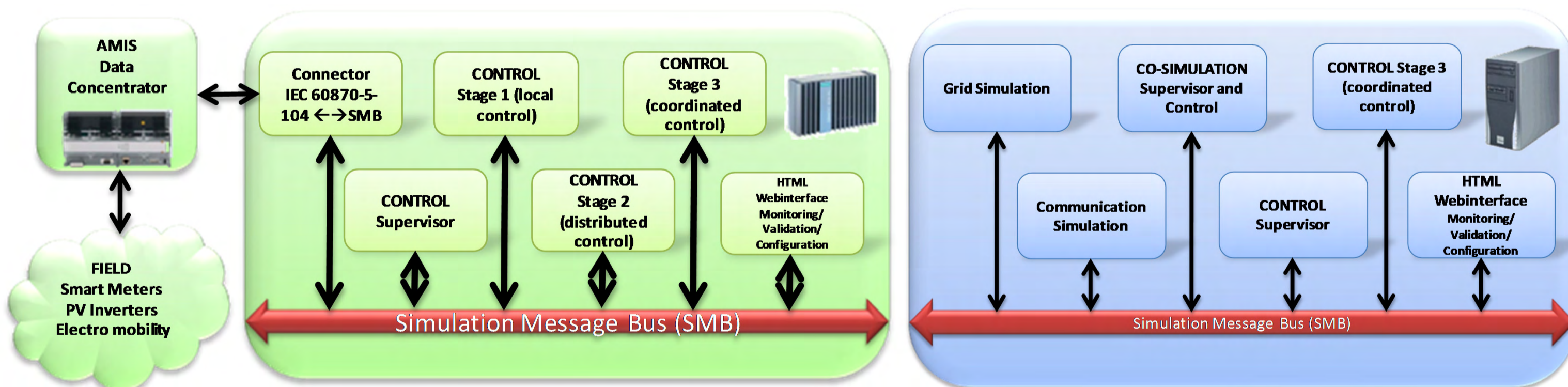


Abb. 2: Zielsystem mit vier Regelstufen

Abb. 3: Co-Simulationssystem

## C-HIL Basisansatz

Abb. 4 zeigt den Basisansatz für die C-HIL-Integration.

- Co-Simulationsumgebung (MS Windows)
  - Strom- und Kommunikationssnetzsim.
  - C-HIL Supervisor & Control
  - Benutzer-Interface (Dashboard)
  - Simulation Message Bus (SMB)
  - Control Supervisor
- Zielumgebung (Ubuntu Linux auf Industrie PC)
  - Regelalgorithmen (Stage 1 – 3)

In dieser Basisintegration werden ausschließlich die drei Regelalgorithmen auf der Feld/Ziel-Hardware (grün) unter Ubuntu-Linux getestet. Alle anderen Komponenten der Co-Simulation, der Simulations-Supervisor, sowie der Control-Supervisor und die Middleware zum Nachrichtenaustausch (SMB) laufen auf dem Simulationsrechner (blau).

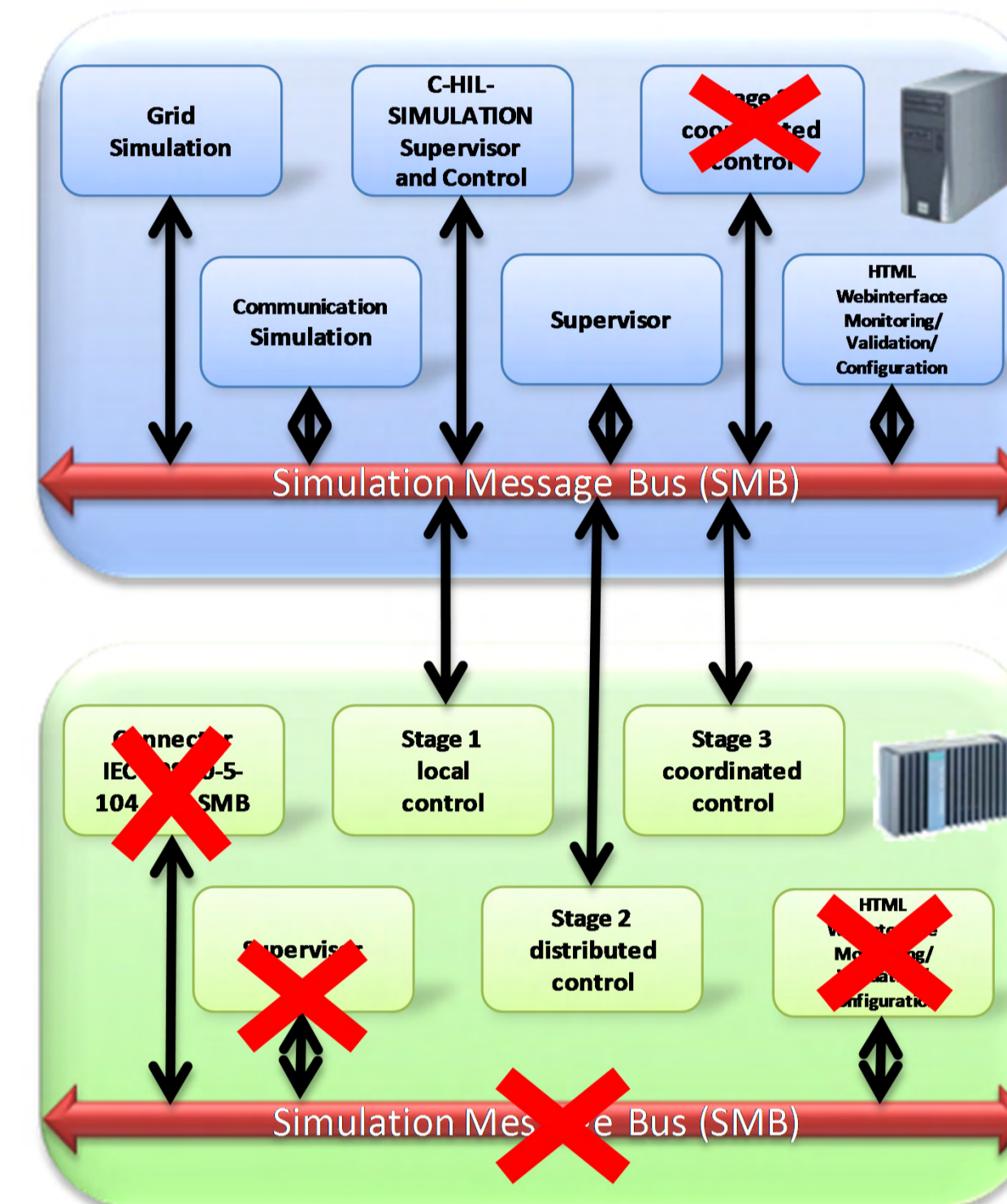


Abb. 4: C-HIL Basisansatz

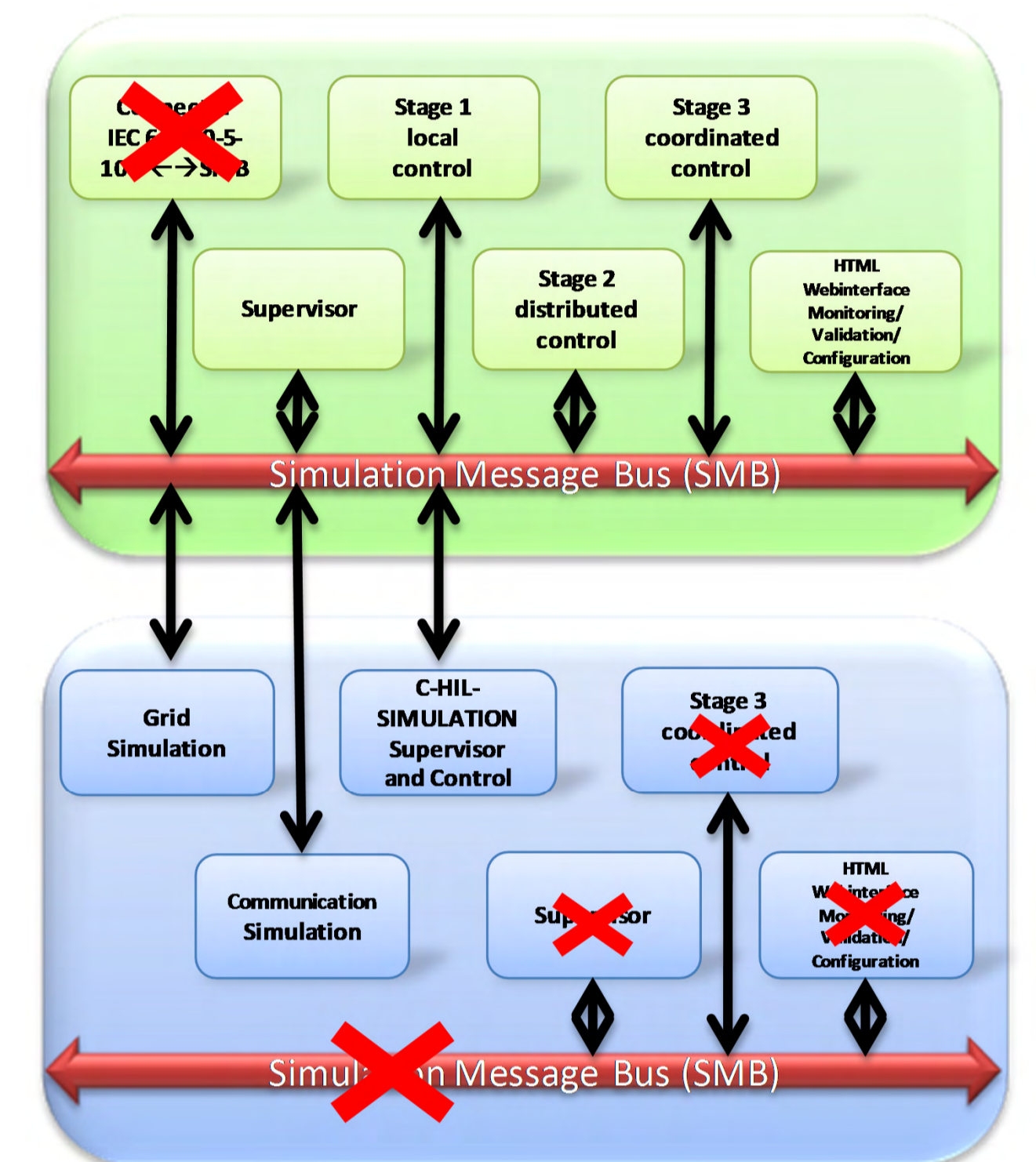


Abb. 5: Feldnaher C-HIL Ansatz

## Feldnaher C-HIL Ansatz

Abb. 5 zeigt den feldnahen C-HIL Ansatz. In diesem Schritt werden nun auch der Control-Supervisor und die Middleware (SMB) auf dem Zielsystem getestet.

- Co-Simulationsumgebung (MS Windows)
  - Strom- und Kommunikationssnetzsim.
  - C-HIL Supervisor & Control
- Zielumgebung (Ubuntu Linux auf Industrie PC)
  - Benutzer-Interface (Dashboard)
  - Simulation Message Bus (SMB)
  - Control Supervisor
  - Regelalgorithmen (Stage 1 – 3)

## Kombinierter C-HIL Ansatz

Dieser Schritt vereint die beiden vorigen Ansätze durch die Kopplung der SMB-Middleware. Das Zielsystem wird wie im Feld betrieben und kann auf das performante Simulationssystem zugreifen. Dies erlaubt eine schnelle, feldnahe Controllerentwicklung und Tests und damit eine Minimierung des Fehlerrisikos (vgl. Abb. 6).

Im Projekt wurde sowohl der Basisansatz, als auch der feldnahe C-HIL Ansatz zur Reglerentwicklung erfolgreich eingesetzt. Der kombinierte Ansatz wird derzeit bei der Weiterentwicklung der SMB-Middleware berücksichtigt.

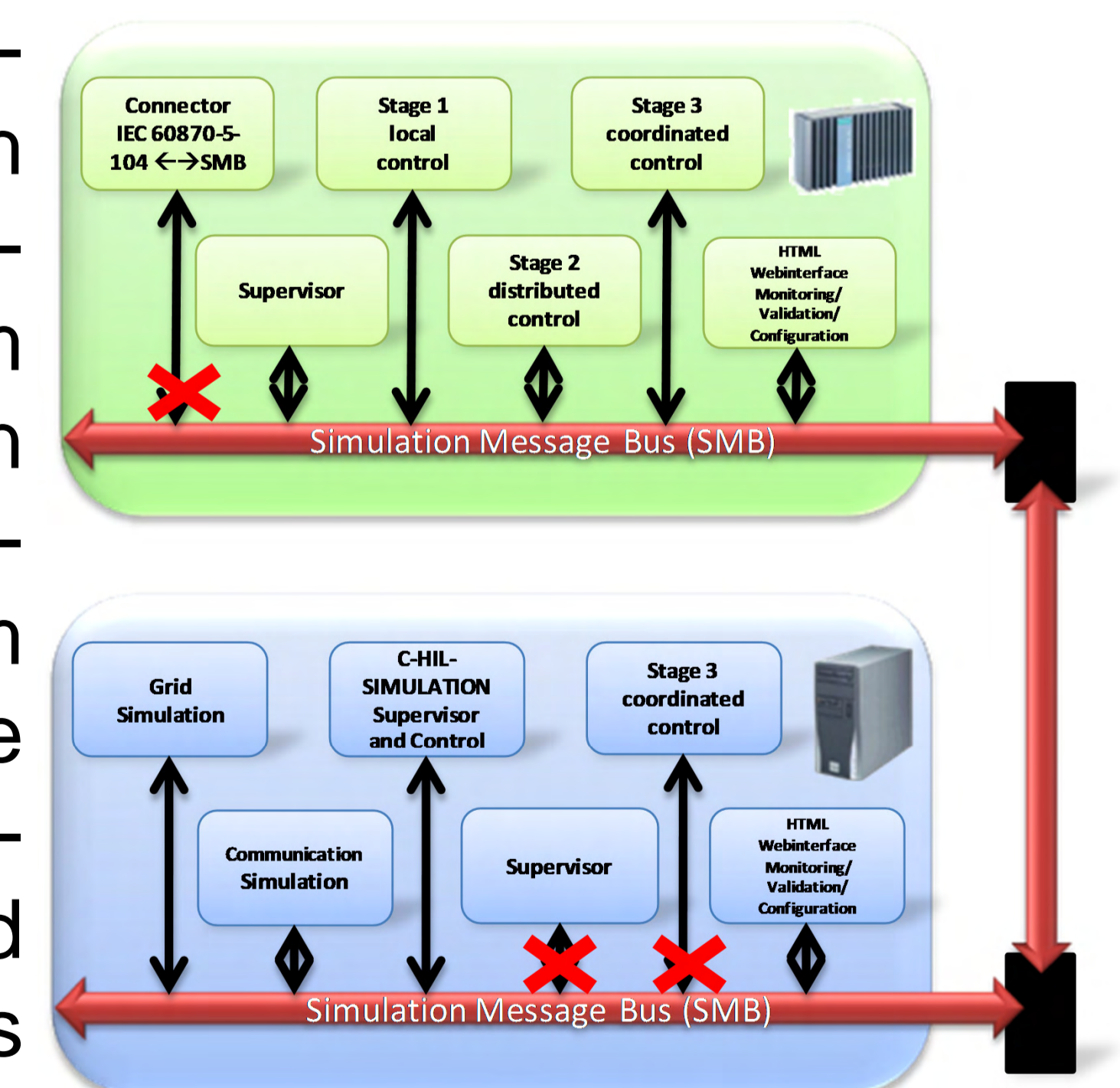


Abb. 6: Kombiniertes C-HIL Ansatz