Abschlussbericht

Verbundprojekt: Mannheim CeCaS – Central Car Server – Supercomputing für Automotive

Teilvorhaben: Einsetzbarkeit dynamischer Softwarearchitekturen in Central Car Servern unter Berücksichtigung von Safety- und Securityanforderungen

Zuwendungsempfänger: Kernkonzept GmbH, Buchenstraße 16b, 01097 Dresden

Förderkennzeichen: 16ME0809

Berichtszeitraum: 01.12.2022 – 30.11.2025

Kurzdarstellung zum Teilvorhaben

Aufgabenstellung

Das Teilvorhaben der Kernkonzept GmbH wurde im Rahmen des Verbundprojekts CeCaS (“Central Car Server – Supercomputing für Automotive”) der vom BMFTR geförderten Initiative MANNHEIM durchgeführt. Übergeordnetes Ziel des Verbundprojekts war die Erforschung und Entwicklung einer Automotive Supercomputing Platform für hochautomatisierte vernetzte Fahrzeuge. Innerhalb des Verbundprojekts hatte Kernkonzept als Hersteller des quelloffenen L4Re-Mikrokern-Betriebssystems die Aufgabe, die Systembasissoftware – insbesondere den L4Re-Hypervisor – an die Anforderungen zukünftiger Central-Car-Server-Architekturen anzupassen. Die Forschungs- und Entwicklungsschwerpunkte umfassten die Untersuchung dynamischer Softwarearchitekturen unter Safety- und Securityanforderungen, die Entwicklung von Werkzeugen zur Konfiguration der Systempartitionierung und deren Eignung für den Nachweis von Safetyanforderungen, die Weiterentwicklung der Cross-Node-Kommunikation, die Anpassung der Schedulingmechanismen sowie die Erweiterung der Testumgebungen.

Voraussetzungen, unter denen das Vorhaben durchgeführt wurde

Kernkonzept ist seit der Unternehmensgründung führend in der Entwicklung des quelloffenen L4Re-Mikrokernsystems aktiv. Wesentliche Vorarbeiten für CeCaS stammen aus den Projekten EMDRIVE (BMBF), Versecloud (BMBF) sowie aus den europäischen Projekten EPI SGA1 und SGA2. Das Projekt wurde als Teilvorhaben in einem Verbundprojekt mit zahlreichen Industrie- und Forschungspartnern durchgeführt; Kernkonzept war primär in den Arbeitspaketen AP1 (Anforderungen und Systemarchitektur), AP2 (Server-Plattform), AP3 (High Performance Prozessor) sowie schwerpunktmäßig AP5 (Softwarearchitektur) tätig.

Planung und Ablauf des Vorhabens

Das Teilvorhaben umfasste eine Laufzeit von 36 Monaten (01.12.2022 bis 30.11.2025). Im ersten Projektjahr standen die Anforderungserhebung sowie der Aufbau der konzeptionellen und infrastrukturellen Grundlagen im Vordergrund. Das zweite Projektjahr verlagerte den Schwerpunkt auf die Umsetzung: Zentrale Mechanismen des Hypervisors wurden überarbeitet, die Test- und

Evaluationsinfrastruktur wurde ausgebaut, und das Design neuer Schedulingmechanismen wurde begonnen. Das dritte Projektjahr stand im Zeichen der Konsolidierung und Integration der entworfenen Mechanismen sowie der Beratung der Projektpartner bei der Demonstratorentwicklung.

Wissenschaftlicher und technischer Stand zu Projektbeginn

Klassische Hypervisorsysteme aus dem Server- und Cloud-Bereich erfüllen die Echtzeit- und Safetyanforderungen des Automobilbereichs nicht; speziell im Automotive-Bereich entwickelte Hypervisoren existieren zwar, sind aber häufig proprietär. Mikrokern-basierte Systeme der dritten Generation wie L4Re bilden eine vielversprechende Grundlage, da sie durch ihre kleine Trusted Computing Base, strenge Capability-basierte Rechteverwaltung und nachweisbare Isolationseigenschaften besonders gut für sicherheitskritische Konsolidierungsszenarien geeignet sind. Eine Anpassung an die spezifischen Anforderungen heterogener Automotive-HPC-Systeme – insbesondere hinsichtlich Cross-Node-Kommunikation, Scheduling unter Mixed-Criticality-Bedingungen und Konfigurationswerkzeugen – stand zu Projektbeginn noch aus.

Zusammenarbeit mit anderen Stellen

Die Zusammenarbeit mit den Projektpartnern erfolgte im Rahmen der Arbeitspakete AP1 bis AP5. Kernkonzept brachte die Sicht des Hypervisors und Betriebssystems in die Anforderungs- und Architekturentwicklung ein und stand den Projektpartnern bei der Demonstratorentwicklung beratend zur Seite. Insbesondere im Bereich der Seitenkanal-Untersuchungen (AP3.3) erfolgte ein intensiver Austausch zwischen Hardware- und Softwarepartnern. Darüber hinaus beteiligte sich Kernkonzept – ohne eigene budgetierte Aufwände in AP9 – an der inhaltlichen Ausgestaltung der Demonstratoren des Verbundprojekts, primär aus Story-Perspektive.

Erzielte Ergebnisse

Im Teilvorhaben wurden die Fähigkeiten des L4Re-Mikrokernsystems substanziell erweitert. Die native Netzwerkunterstützung wurde um ein generisches Framework ausgebaut und durch reproduzierbare Performancemessungen in der Continuous-Integration-Infrastruktur abgesichert. Das DRQ-Subsystem wurde überarbeitet, sodass eine Beeinflussung des temporalen Verhaltens zwischen Partitionen auch bei timeout-behafteten IPC-Kanälen ausgeschlossen wird. Erweiterte Schedulingkontexte wurden als first-class citizens im Betriebssystemkern verankert, und ein tickless Idle-Modus wurde in den Open-Source-Hauptentwicklungszweig integriert. Ein

prototypisches Konfigurationswerkzeug für den Hypervisor sowie Designkonzepte für die Integration von Safety-Checks wurden erarbeitet. Die Test- und Safety-Infrastruktur wurde umfassend ausgebaut.

Die wissenschaftlichen Erkenntnisse aus den Schedulingerweiterungen werden in der Veröffentlichung “Scheduling Constraints: A Universal OS Mechanism for Managing Shared Resources” auf der RTAS 2026 erscheinen.

Verwertung

Die im Projekt entwickelten Werkzeuge zur Konfiguration und Partitionierung des Hypervisors sollen unmittelbar nach Projektabschluss als lizenzierbare Produkte angeboten werden. Die Erweiterungen des Open-Source-Hypervisorsystems werden über die regulären Releases öffentlich zugänglich gemacht; entsprechende Folgeprodukte und Dienstleistungsangebote werden zwei bis vier Jahre nach Projektabschluss erwartet.

Eingehende Darstellung zum Teilvorhaben

Inhaltsverzeichnis

- 2.1 Aufgabenstellung und wissenschaftlicher Stand
  - Aufgabenstellung des Teilvorhabens
  - Wissenschaftlicher und technischer Stand zu Projektbeginn
  - Eigene Vorarbeiten und Anknüpfungspunkte
- 2.2 Ablauf des Vorhabens
  - 2.2.1 Projektlaufzeit
  - 2.2.2 Chronologischer Ablauf
    - Erstes Projektjahr (12/2022 – 11/2023)
    - Zweites Projektjahr (12/2023 – 11/2024)
    - Drittes Projektjahr (12/2024 – 11/2025)
  - 2.2.3 Abweichungen von der ursprünglichen Planung
- 2.3 Wesentliche Ergebnisse
  - Cross-Node-Kommunikation und natives Networking
  - Schedulingmechanismen für Mixed-Criticality-Systeme
  - Konfigurationstooling für den Hypervisor
  - Test- und Safety-Infrastruktur
  - Beiträge zur Anforderungs- und Architekturentwicklung
  - Integration mit den Demonstratoren des Verbundprojekts
- 2.4 Notwendigkeit und Angemessenheit der geleisteten Projektarbeiten
- 2.5 Voraussichtlicher Nutzen im Sinne des fortgeschriebenen Verwertungsplans
  - 2.5.1 Erfindungen, Schutzrechtsanmeldungen und erteilte Schutzrechte
  - 2.5.2 Wirtschaftliche Erfolgsaussichten nach Projektende
  - 2.5.3 Wissenschaftliche und technische Erfolgsaussichten nach Projektende
  - 2.5.4 Wissenschaftliche und wirtschaftliche Anschlussfähigkeit
- 2.6 Dem Zuwendungsempfänger bekannt gewordener Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen
- 2.7 Erfolgte oder geplante Veröffentlichungen des Ergebnisses

2.1 Aufgabenstellung und wissenschaftlicher Stand

Aufgabenstellung des Teilvorhabens

Das Teilvorhaben der Kernkonzept GmbH wurde im Rahmen des Verbundprojekts CeCaS (“Central Car Server – Supercomputing für Automotive”) der vom BMFTR geförderten Initiative MANNHEIM durchgeführt. Das übergeordnete Ziel des Verbundprojekts war die Erforschung und Entwicklung einer Automotive Supercomputing Platform – eines leistungsfähigen Central-Car-Server-Konzepts auf Basis neuer Automotive-qualifizierter Hochleistungsprozessoren in FinFET-Technologie, unterstützt durch applikationsspezifische Beschleuniger und einen adaptiven Automotive-Software-Stack für hochautomatisierte vernetzte Fahrzeuge.

Innerhalb des Verbundprojekts hatte Kernkonzept als Hersteller des quelloffenen L4Re-Mikrokern-Betriebssystems die Aufgabe, die Systembasissoftware – insbesondere den L4Re-Hypervisor – an die Anforderungen zukünftiger Central-Car-Server-Architekturen anzupassen und weiterzuentwickeln. Die Forschungs- und Entwicklungsziele des Teilvorhabens umfassten die Untersuchung der Einsetzbarkeit dynamischer Softwarearchitekturen unter Safety- und Securityanforderungen, die Entwicklung von Tools zur Konfiguration der Systempartitionierung und deren Untersuchung auf Eignung zum Nachweis der Safetyanforderungen, die Weiterentwicklung von Mechanismen zur Cross-Node-Kommunikation, die Anpassung der Schedulingmechanismen an die Anforderungen des Central Car Servers sowie die Erweiterung der Testumgebungen für komplexe Car-Server-Testszenarien.

Diese Schwerpunkte wurden in den Arbeitspaketen AP1 (Anforderungen und Systemarchitektur), AP2 (Server-Plattform), AP3 (High Performance Prozessor) sowie schwerpunktmäßig AP5 (Softwarearchitektur) bearbeitet und tragen primär zu den Verbundzielen 4 (HW/SW-Codesign inklusive Betriebssystemintegration), 8 (ASIL-D auf Systemebene) und 9 (Situationsgerechter EE-Bordnetzbetrieb) bei.

Wissenschaftlicher und technischer Stand zu Projektbeginn

Die Automobilelektronik befindet sich in einem tiefgreifenden Wandel: Die klassische, domänenorientierte E/E-Architektur mit einer Vielzahl spezialisierter Steuergeräte stößt angesichts der wachsenden Software-Komplexität – insbesondere im Kontext des automatisierten Fahrens, der Elektrifizierung und der Vernetzung – zunehmend an ihre Grenzen. Aktuelle Architekturtrends gehen daher in Richtung einer stärkeren Zentralisierung der Rechenleistung in wenigen

leistungsfähigen Zentralrechnern (Central Car Server). Bisherige verfügbare Lösungen für solche Zentralrechner basieren entweder auf Prozessoren aus dem Consumer-Bereich oder auf der Kombination leistungsfähiger SoCs mit zusätzlich angekoppelten klassischen Automotive-Mikrocontrollern; eine durchgängige ASIL-D-Qualifizierung auf Systemebene bei gleichzeitig hoher Rechenleistung ist mit beiden Ansätzen nicht erreichbar.

Aus Sicht der Systembasissoftware ergeben sich aus dieser Entwicklung mehrere Herausforderungen, die zu Projektbeginn nur unvollständig adressiert waren. Klassische Hypervisorsysteme aus dem Server- und Cloud-Bereich erfüllen die Echtzeit- und Safetyanforderungen des Automobilbereichs nicht; speziell im Automotive-Bereich entwickelte Hypervisoren existieren zwar, sind aber häufig proprietär und auf konkrete Zielplattformen zugeschnitten. Mikrokern-basierte Systeme der dritten Generation wie L4Re bilden eine vielversprechende Grundlage, da sie durch ihre kleine Trusted Computing Base, strenge Capability-basierte Rechteverwaltung und nachweisbare Isolationseigenschaften besonders gut für sicherheitskritische Konsolidierungsszenarien geeignet sind. Eine Anpassung an die spezifischen Anforderungen heterogener Automotive-HPC-Systeme – insbesondere hinsichtlich Cross-Node-Kommunikation, Scheduling unter Mixed-Criticality-Bedingungen und Konfigurationswerkzeugen – stand zu Projektbeginn jedoch noch aus.

Mit der zunehmenden Komplexität der zu konsolidierenden Anwendungen und der unterliegenden Hardware steigt auch die Komplexität der notwendigen Systemkonfiguration. Bestehende Werkzeuge bieten wenig Unterstützung bei der visuellen Darstellung der Systempartitionierung oder bei der automatisierten Prüfung gegen Safetyanforderungen. Klassische Schedulingmechanismen wiederum basieren überwiegend auf der Annahme, dass die CPU-Zeit die einzige knappe und zu verteilende Ressource ist; in heterogenen HPC-Systemen sind jedoch weitere geteilte Ressourcen wie Speicherbusbandbreite oder Cache-Kapazität ebenfalls für die Sicherstellung von Freedom from Temporal Interference relevant. Ansätze, die solche Ressourcen als first-class citizens im Scheduling berücksichtigen, waren zu Projektbeginn Gegenstand laufender Forschung, jedoch noch nicht in produktionsreifen Hypervisorsystemen umgesetzt. Auch die native Treiber- und Netzwerkstack-Unterstützung für Cross-Node-Kommunikation sowie die Integration von Requirements Tracing und Coverage Reporting in den Continuous-Integration-Workflow waren in L4Re zu Projektbeginn nur eingeschränkt verfügbar.

Eigene Vorarbeiten und Anknüpfungspunkte

Kernkonzept ist seit der Unternehmensgründung führend in der Entwicklung des quelloffenen L4Re-Mikrokernsystems aktiv, dessen Wurzeln bis zur Entwicklung der L4-Mikrokernfamilie an der TU Dresden ab 1997 zurückreichen. Das System wird bereits erfolgreich in Anwendungsbereichen mit hohen Sicherheitsanforderungen eingesetzt; eine Common-Criteria-EAL4+-Zertifizierung wurde im Februar 2025 erteilt. Eine Safetyzertifizierung nach ISO 26262 wurde im August 2024 durch den TÜV Süd erteilt; die Zertifizierung erfolgte außerhalb des CeCaS-Projekts und wurde gemeinsam mit Elektrobit unter dem Produktnamen “EB corbos Hypervisor” durchgeführt.

Wesentliche Vorarbeiten für CeCaS stammen aus den vom BMBF bzw. BMFTR geförderten Projekten EMDRIVE (Real-Time-kompatible Erweiterung von Zentralrechenplattformen mit Fokus auf dynamischen Betriebsstrategien) und Versecloud (sichere und nachweisbare Virtualisierungsgrundlagen unter Einsatz formaler Methoden) sowie aus den europäischen Projekten EPI SGA1 und SGA2 (European Processor Initiative). Die in diesen Projekten entwickelten Mechanismen bildeten die Grundlage, auf der die in CeCaS adressierten Erweiterungen für den Einsatz in Central-Car-Server-Architekturen aufbauen konnten. Die Veröffentlichung der Ergebnisse als Open Source leistet einen Beitrag zur technologischen Souveränität Deutschlands und Europas im Bereich automotive-tauglicher, sicherheitszertifizierbarer Systembasissoftware.

2.2 Ablauf des Vorhabens

2.2.1 Projektlaufzeit

Die Laufzeit des Teilvorhabens erstreckte sich vom 01.12.2022 bis zum 30.11.2025 und umfasste damit 36 Projektmonate. Diese Laufzeit entspricht der ursprünglichen Planung gemäß der Teilvorhabensbeschreibung.

2.2.2 Chronologischer Ablauf

Das Teilvorhaben von Kernkonzept umfasste über die gesamte Projektlaufzeit von 36 Monaten Beiträge zu den Arbeitspaketen AP1 (Anforderungen und Systemarchitektur), AP2 (Server-Plattform), AP3 (High Performance Prozessor) sowie als Arbeitsschwerpunkt AP5 (Softwarearchitektur). Im Folgenden werden die wesentlichen Arbeiten und Ergebnisse chronologisch nach Projektjahren dargestellt.

Erstes Projektjahr (12/2022 – 11/2023)

Das erste Projektjahr stand im Zeichen der Anforderungserhebung und der Schaffung der konzeptionellen und infrastrukturellen Grundlagen für die in den folgenden Projektjahren geplanten Entwicklungsarbeiten.

Anforderungen für Central-Car-Server-Architekturen (AP1, AP2, AP3, AP5.1). Kernkonzept beteiligte sich aktiv an der Sammlung der Anforderungen für die im Projekt zu erforschende Central-Car-Server-Architektur. Einerseits wurden Anforderungen an die Hardware aus Sicht des Hypervisors formuliert. Dabei handelt es sich um grundlegend notwendige Eigenschaften, die zur Sicherstellung der Isolation von Nutzlasten erforderlich sind. Die Erfüllung dieser Anforderungen durch die Hardware ist die Grundlage dafür, dass der Hypervisor Safetyanforderungen wie “freedom from spatial interference” und “freedom from temporal interference” auf einem High-Performance-Knoten der CeCaS-Architektur durchsetzen kann. Ebenso sind sie grundlegend für die Einhaltung der Sicherheitsziele der Nutzlasten (Cybersecurity). Diese Anforderungen sind in den im Rahmen von AP1 ausgearbeiteten Anforderungskatalog und die Architekturspezifikation (Meilensteine M1.1 und M1.2) eingeflossen.

Des Weiteren nahm Kernkonzept die Anforderungen der Projektpartner hinsichtlich der Eigenschaften an die Systembasissoftware auf. Diese Anforderungen betrafen primär die Isolation bzw. das Weiterreichen spezieller Hardware (zum Beispiel KI-Beschleuniger) an voneinander isolierte Partitionen des Systems (zum Beispiel Container, virtuelle Maschinen oder hypervisor-native Anwendungen). Dabei wurde sowohl mit den Partnern aus AP1 als auch AP2 zusammengearbeitet, um die Anforderungen aus Software- und Hardwaresicht zu ermitteln und zu berücksichtigen.

Die Sammlung dieser Anforderungen diente zudem als Grundlage für die Entwicklung der Prototypen der Integrationstools für Systemintegratoren. Hier wurde ein Fokus auf die einfache und transparente Partitionierung des Systems mittels intuitiver Nutzerschnittstellen gelegt – eine fundamentale Voraussetzung für die Konzeption und Entwicklung der Konfigurationsdatenbank und -tools in AP5.3.4, deren Arbeiten im ersten Projektjahr begonnen wurden.

Test- und Safety-Infrastruktur (AP5.2, AP5.3.1, AP5.3.9). Unter Berücksichtigung der vorausgegangenen Arbeiten zu MCU-Systemen, welche Grundlage der Cross-Node-Kommunikation sind, wurde mit dem Aufbau einer Test- und Benchmarkinfrastruktur begonnen. Diese dient dazu, die Performanceauswirkungen der Entwicklungen zur Cross-Node-Kommunikation

und der neuen Schedulingmechanismen zuverlässig evaluieren zu können – eine Voraussetzung für die im weiteren Projektverlauf in AP5.3.9 vorgesehenen Evaluationsarbeiten. Außerdem wurde die Integration des Safety-Toolings (Requirements Tracing, Coverage Reporting) in den allgemeinen Testworkflow verbessert und in Teilen bereits als Open Source veröffentlicht (Kernel-Coverage-Werkzeuge und deren Integration). Diese Integration ermöglicht den kontinuierlichen Nachweis der Sicherheitseigenschaften des Systems.

Konfigurationstooling für den Hypervisor (AP5.3.4, AP5.3.5). Unter Einbeziehung der Erkenntnisse aus der Anforderungserhebung wurde mit der prototypischen Implementierung eines Konfigurationstools begonnen. Dabei wurde besonderer Wert auf die Definition und Spezifikation der Schnittstellen der einzelnen Komponenten des Hypervisors gelegt.

Aufgrund der vielen Freiheitsgrade eines HPC-Systems im Automobilbereich stellt die Konfiguration und Partitionierung des Hypervisorsystems eine besondere Herausforderung für Systemintegratoren dar. So kann zu nahezu beliebigen Zeitpunkten im Entwicklungsprozess des Gesamtsystems die Platzierung von Komponenten zwischen Realtime-Cores (MPU-basiert) und High-Performance-Cores (MMU-basiert) ohne zusätzlichen Programmieraufwand geändert werden. Es ist daher wichtig, dass Systemintegratoren Werkzeuge zur Verfügung stehen, die diese Aufgabe so einfach wie möglich gestalten, dabei aber auch die Einhaltung von Safety- und Securityeigenschaften sichtbar machen und – soweit möglich – sogar automatisiert überwachen können.

Die Konfiguration muss dabei automatisiert sowohl für große HPC-Systeme aus dem Automobilbereich (dynamische Lua-Skripte), Safety-enabled HPC-Systeme (ASIL-B; Compile-time-Konfiguration) als auch für MPU-Systeme (ASIL-D; größenoptimierte tinit-Konfiguration) generiert werden können. Das Design des Werkzeugs wurde so konzeptioniert, dass es sowohl zur Konfiguration von virtuellen Maschinen und Containern als auch für native Anwendungen geeignet ist. Da das L4Re-Hypervisorsystem ein modernes, mikrokernbasiertes Capability-System ist, wurde besonderer Wert auf die übersichtliche Darstellung und Dokumentation der Capability-Verteilung gelegt. Diese ist eine Voraussetzung für den später im Projektverlauf zu entwickelnden Safety Configuration Checker (AP5.3.5) und ermöglicht die Partitionierung des Systems zur Einhaltung der Safety- und Securityziele (temporale und räumliche Freedom from Interference sowie Cybersecurity).

Zweites Projektjahr (12/2023 – 11/2024)

Im zweiten Projektjahr verlagerte sich der Schwerpunkt von der Konzeption auf die Umsetzung. Zentrale Mechanismen des Hypervisors wurden überarbeitet, die Test- und Evaluationsinfrastruktur wurde ausgebaut, und mit dem Design neuer Schedulingmechanismen wurde begonnen.

Anforderungen für Central-Car-Server-Architekturen (AP1, AP2, AP3, AP5.1). Die Definition des Anforderungskatalogs im Rahmen von AP1 und AP2.1 wurde im vorherigen Berichtsjahr abgeschlossen. Kernkonzept folgte jedoch weiterhin den Diskussionen in AP1, in deren Rahmen der Anforderungskatalog kontinuierlich auf Übereinstimmung mit den Entwicklungen im Gesamtprojekt überprüft wurde. Wir lieferten dabei die Sicht des Betriebssystems und Hypervisors als Input für Diskussionen. Der Hauptfokus in AP2.2 lag auf der Definition des Server-Schnittstellen-Stecksystems und der entsprechenden Module sowie der Kommunikationsnetze. Bei der Definition der einzubettenden Recheneinheiten lieferte Kernkonzept einerseits Input hinsichtlich der Optionen aus Sicht von Betriebssystem und Hypervisor. Andererseits flossen Erkenntnisse aus der Komponentenauswahl und den Designentscheidungen in AP2 direkt in die Arbeiten von Kernkonzept im Hauptarbeitspaket AP5 zurück. In AP3 beobachtete Kernkonzept die Entwicklung der neuen Hardware und unterstützte durch seine Expertise im Betriebssystembereich bei Designentscheidungen hinsichtlich der Verwendbarkeit neuer Features in zertifizierbar funktional-sicheren Betriebssystemen und Hypervisoren. In AP5.1 wurde im zweiten Projektjahr in Zusammenarbeit mit den Projektpartnern die Anforderungsdefinition an die Software abgeschlossen.

Test- und Safety-Infrastruktur (AP5.2). Die Testinfrastruktur für die neuen Mechanismen wurde im zweiten Projektjahr weiter verbessert. Dabei wurden insbesondere Grundlagen für die kontinuierliche Evaluation der Leistungsparameter des Systems im Rahmen der Continuous Integration gelegt. Weiterhin wurde die Infrastruktur geschaffen, damit auch Userland-Komponenten Coverage-Informationen im Continuous Testing bereitstellen können und diese Daten ausgewertet werden können.

Cross-Node Communication (AP5.3.2). Basierend auf den vorangegangenen Arbeiten wurden die Fähigkeiten des L4Re-Betriebssystems für Cross-Node-Kommunikation verbessert. Der Fokus lag im zweiten Projektjahr vor allem auf einem verbesserten Ethernet-Support, damit auch native Treiber und Netzwerkstacks zur Verfügung gestellt werden können. Weiterhin wurde der

Support für die entsprechenden POSIX-Schnittstellen zur Anbindung der Netzwerkhardware verbessert.

Konfigurationstooling für den Hypervisor (AP5.3.6). Das Tooling für das Anforderungstracing wurde um Mechanismen erweitert, welche die Safetyeigenschaften innerhalb des Systems korrekt zu den entsprechenden Requirements und deren Tests tracen können. Dabei wurde eine Integration in das Continuous Testing erreicht. Beim Integrationstooling wurden erste Designkonzepte erarbeitet, wie eine Integration von Safety-Checks in den Konfigurator ermöglicht werden kann. Die konkrete Umsetzung stellte sich jedoch als sehr komplex heraus, sodass hier weiterhin signifikante Arbeiten am Konzept notwendig waren. Dies betraf einerseits die Abbildung der Regeln im Framework, andererseits aber auch die Möglichkeiten zur visuellen Darstellung der Safetypartitionen. Hier wurden im folgenden Projektjahr weitere Lösungen diskutiert und erarbeitet.

Design neuer Schedulingmechanismen (AP5.3.7). Im zweiten Projektjahr wurde im Rahmen der Arbeiten zu AP5 mit dem Design der neuen Schedulingmechanismen begonnen. Dabei gab es mehrere Fokuspunkte:

Eine Überarbeitung des DRQ-Subsystems zur asynchronen Kommunikation zwischen Cores im Kern wurde konzipiert. Hier müssen, zur Erreichung besserer Freedom-from-Interference-Eigenschaften, die Mechanismen so überarbeitet werden, dass eine Beeinflussung des temporalen Verhaltens zwischen Partitionen auch bei der Nutzung von timeout-behafteten IPC-Kanälen ausgeschlossen werden kann. Es wurde ein Design erarbeitet, dessen Umsetzung für das dritte Projektjahr vorgesehen war.

Außerdem wurde mit der Entwicklung erweiterter Schedulingkontexte begonnen. Eine Softwareanpassung, welche Schedulingkontexte als first-class citizens im Betriebssystemkern verankert, wurde entworfen und prototypisch implementiert. Dies dient vor allem dazu, Scheduling auch unter Berücksichtigung weiterer Ressourcen wie zum Beispiel Speicherbusbandbreite vornehmen zu können. Die prototypische Implementierung zeigte vielversprechende Ergebnisse; die Integration in den Hauptentwicklungszweig benötigte jedoch noch weitere Arbeit.

Schließlich wurde ein tickless Idle-Modus konzipiert. Dieser Mechanismus ist ein erster Schritt, um Interferenz durch Timer-Ticks in Echtzeitszenarien zu reduzieren oder zu eliminieren. Das erste Design war dabei primär auf das

Energiesparen ausgerichtet, um unnötiges Aufwachen von Cores zu verhindern. Die Implementierung des Konzepts war für das dritte Projektjahr vorgesehen.

Drittes Projektjahr (12/2024 – 11/2025)

Das dritte Projektjahr stand im Zeichen der Konsolidierung und Integration der in den Vorjahren entworfenen Mechanismen. Mehrere zentrale Entwicklungen wurden zur Reife gebracht und in den Hauptentwicklungszweig des L4Re-Systems überführt. Darüber hinaus erfolgte die Integration der Kernkonzept-Beiträge in die Demonstratorenarbeit der Projektpartner.

Beratung zur sicheren Systemarchitektur in Demonstratoren (AP2.2, AP3.3). Im dritten Projektjahr legte Kernkonzept einen Schwerpunkt auf die Beratung der Projektpartner bei der Demonstratorentwicklung im Hinblick auf eine konsistente und sichere Architektur aus Sicht von Hypervisor und Betriebssystem. Es wurden Empfehlungen gegeben, an welchen Stellen der Architektur der Einsatz eines Hypervisors sinnvoll ist und welche Aspekte bei der Auswahl von Hardware und weiterer Software zu berücksichtigen sind, um die Komponenten im Sinne von Safety und Security sicher betreiben zu können. Speziell im Rahmen von AP3.3 lieferte Kernkonzept Input zu möglichen, zu vermeidenden Seitenkanälen und untersuchte die von den Projektpartnern betrachteten Seitenkanäle auf ihre Anwendbarkeit auf den Hypervisor. Eine Härtung des Hypervisors selbst war auf Basis dieser Untersuchungen nicht erforderlich; die Erkenntnisse fließen jedoch in die Security Guidance für den sicheren Einsatz des Hypervisors ein.

Cross-Node Communication und Test-Infrastruktur (AP5.3.2, AP5.3.3). Die native Netzwerkunterstützung in L4Re wurde im dritten Projektjahr weiter ausgebaut, wobei besonderer Wert auf ein generisches Framework gelegt wurde. Der entstandene Prototyp wurde umfassend auf seine Performance untersucht. Es wurden Szenariotests entwickelt, um die Mechanismen im Rahmen der Continuous Integration kontinuierlich zu evaluieren. Im Rahmen von AP5.3.3 wurde besonderer Wert auf reproduzierbare Performancemessungen gelegt, deren Ergebnisse über ein Dashboard visualisiert werden. Diese Messungen sind fester Bestandteil der CI-Infrastruktur bei Kernkonzept und stellen sicher, dass sich keine Performance-Regressions in den Hauptentwicklungszweig einschleichen können.

Konfigurationstooling für den Hypervisor (AP5.3.6). Aufbauend auf den Designkonzepten des zweiten Projektjahrs wurde die Darstellung von Safetyeigenschaften über visuelle Hints im Konfigurationstool weiter

konzeptioniert. Eine vollständige prototypische Umsetzung war im verbleibenden Projektbudget nicht mehr realisierbar. Die im Projekt gewonnenen Konzepte und Erkenntnisse fließen jedoch in die weitere Entwicklung des Konfigurationstoolings außerhalb des Projektkontexts ein und bilden dort eine wertvolle Vorarbeit für eine spätere Umsetzung.

Schedulingmechanismen (AP5.3.8, AP5.3.9). Die im zweiten Projektjahr begonnene Überarbeitung des DRQ-Subsystems wurde im dritten Projektjahr abgeschlossen. Es wurden neue Tests entwickelt und bestehende Tests erweitert, um die korrekte Umsetzung zu verifizieren und auf Performance-Regressionen zu untersuchen. Wie erwartet wurde nur ein minimaler Performanceverlust in wenigen Corner Cases festgestellt; viele Use Cases profitieren sogar vom eager transfer der IPC-Nachrichten.

Die Arbeiten an den Schedulingkontexten wurden weiter ausgebaut und für die Integration in den Hauptentwicklungszweig vorbereitet. Die wissenschaftlichen Erkenntnisse aus dieser Arbeit wurden in einer Veröffentlichung mit dem Titel “Scheduling Constraints: A Universal OS Mechanism for Managing Shared Resources” zusammengefasst, die auf der RTAS 2026 erscheinen wird.

Die Implementierung des im zweiten Projektjahr begonnenen tickless Idle-Modus wurde finalisiert und ist mittlerweile in den Open-Source-Entwicklungszweig des L4Re Mikrokerns Fiasco integriert und dort öffentlich zugänglich.

Integration mit den Ergebnissen der Projektpartner (AP5.4). Die Integration der Kernkonzept-Beiträge mit den Ergebnissen der Projektpartner wurde im dritten Projektjahr umgesetzt. Zunächst wurde der Hypervisor für die ursprünglich geplante Arm-Neoverse-Plattform verfügbar gemacht. Im Projektverlauf entschieden sich die für die Demonstratoren verantwortlichen Projektpartner jedoch für eine x86-Architektur, die von L4Re ebenfalls unterstützt wird. Bei der konkreten Hardwareauswahl wurde von den Partnern eine AMD-CPU ausgewählt; die Virtualisierungsmechanismen von AMD-Prozessoren (AMD-V/SVM) werden – im Gegensatz zu den entsprechenden Mechanismen von Intel-Prozessoren (Intel VT-x/VMX) – derzeit nicht von L4Re unterstützt. Die Ergebnisse von Kernkonzept mit Virtualisierungsbezug konnten daher nicht in den gemeinsamen Demonstratoren der Projektpartner gezeigt werden, sondern wurden in eigenen Demonstratoren von Kernkonzept evaluiert und präsentiert.

Beiträge zur Demonstrator-Story (AP9). Obwohl Kernkonzept im Arbeitspaket AP9 keine eigenen Aufwände eingeplant hatte, beteiligte sich Kernkonzept im

dritten Projektjahr an der inhaltlichen Ausgestaltung der Demonstratoren des Verbundprojekts. Der Beitrag erfolgte primär aus Story-Perspektive – durch die Einbringung der Hypervisor- und Betriebssystemsicht in die Demonstratorerzählung sowie durch die Abstimmung mit den Projektpartnern zur Darstellung der gemeinsam erarbeiteten Lösungen.

2.2.3 Abweichungen von der ursprünglichen Planung

Wesentliche Abweichungen von der ursprünglichen Arbeits-, Zeit- und Ausgabenplanung haben sich im Projektverlauf nicht ergeben. Die geplanten Meilensteine wurden im Wesentlichen wie vorgesehen erreicht; das eingesetzte Budget bewegte sich im Rahmen der ursprünglichen Planung, Verschiebungen des Budgets zwischen den Teilarbeitspaketen waren nicht erforderlich.

Im Hinblick auf den gemeinsamen Demonstrator des Verbundprojekts ist anzumerken, dass im Projektverlauf eine x86-Plattform mit AMD-Hardware ausgewählt wurde. Da die Virtualisierungsmechanismen von AMD-Prozessoren (AMD-V/SVM) – im Gegensatz zu den entsprechenden Mechanismen von Intel-Prozessoren (Intel VT-x/VMX) – derzeit nicht von L4Re unterstützt werden, konnten die Beiträge von Kernkonzept mit Virtualisierungsbezug im Verbund-Demonstrator nicht gezeigt werden. Auf den von Kernkonzept unterstützten Architekturen (x86_64 und arm64) wurden die Ergebnisse jedoch in verschiedenen Formaten demonstriert.

Insgesamt haben sich die Aussichten für die Erreichung der Projektziele gegenüber dem ursprünglichen Antrag nicht negativ verändert. Eine Änderung der Zielsetzung war nicht erforderlich.

2.3 Wesentliche Ergebnisse

Im Folgenden werden die wesentlichen Ergebnisse des Teilvorhabens nach den Forschungsschwerpunkten gegliedert dargestellt. Eine chronologische Darstellung des Projektverlaufs findet sich in Abschnitt 2.2.2.

Cross-Node-Kommunikation und natives Networking

Im Projekt wurden die Fähigkeiten des L4Re-Betriebssystems zur Cross-Node-Kommunikation in heterogenen Automotive-HPC-Systemen substanziell erweitert. Die native Netzwerkunterstützung wurde um ein generisches Framework ausgebaut, das den Einsatz nativer Treiber und Netzwerkstacks innerhalb von L4Re ermöglicht. Die zugehörigen POSIX-Schnittstellen wurden entsprechend ergänzt, um die Anbindung von Netzwerkhardware durch Standardanwendungen zu vereinfachen.

Die entstandenen Mechanismen wurden umfassend auf ihre Performance untersucht und sind durch eine Sammlung von Szenariotests in der Continuous-Integration-Infrastruktur von Kernkonzept abgesichert. Reproduzierbare Performancemessungen werden über ein Dashboard visualisiert; Regressionen können dadurch unmittelbar erkannt und behoben werden.

Schedulingmechanismen für Mixed-Criticality-Systeme

Das Scheduling-Subsystem des L4Re-Mikrokerns wurde an mehreren Stellen erweitert, um den Anforderungen heterogener Central-Car-Server-Systeme zu genügen.

Eine zentrale Verbesserung betrifft das DRQ-Subsystem für die asynchrone Kommunikation zwischen Cores im Kern. Die Mechanismen wurden überarbeitet, sodass eine Beeinflussung des temporalen Verhaltens zwischen Partitionen auch bei der Nutzung von timeout-behafteten IPC-Kanälen ausgeschlossen wird. Damit wird eine wichtige Voraussetzung für die Sicherstellung von Freedom from Temporal Interference geschaffen. Performance-Messungen zeigen, dass die Überarbeitung in den meisten Anwendungsszenarien sogar zu einer Verbesserung des Laufzeitverhaltens führt; lediglich in wenigen Corner Cases wurde ein minimaler Performanceverlust festgestellt.

Ein zweiter Schwerpunkt war die Entwicklung erweiterter Schedulingkontexte. Diese wurden als first-class citizens im Betriebssystemkern verankert und ermöglichen es, Schedulingentscheidungen unter Berücksichtigung weiterer Ressourcen wie Speicherbusbandbreite zu treffen – ein Aspekt, der in klassischen Schedulingverfahren nicht abgedeckt ist, in heterogenen HPC-Systemen jedoch zunehmend an Bedeutung gewinnt. Die wissenschaftlichen Erkenntnisse aus dieser Arbeit wurden in einer Veröffentlichung mit dem Titel “Scheduling Constraints: A Universal OS Mechanism for Managing Shared Resources” zusammengefasst, die auf der RTAS 2026 erscheinen wird.

Schließlich wurde ein tickless Idle-Modus konzipiert, implementiert und in den Open-Source-Hauptentwicklungszweig des L4Re-Mikrokerns Fiasco integriert. Dieser Mechanismus reduziert unnötige Aufwachvorgänge von Cores und stellt einen ersten Schritt zur Reduktion von Interferenzen durch Timer-Ticks in Echtzeitszenarien dar.

Konfigurationstooling für den Hypervisor

Im Projektverlauf wurde ein Konfigurationswerkzeug für den L4Re-Hypervisor prototypisch entwickelt, das die Partitionierung des Systems durch Systemintegratoren erleichtert. Kernpunkte des Werkzeugdesigns sind die übersichtliche Darstellung und Dokumentation der Capability-Verteilung sowie die Unterstützung verschiedener Konfigurationsformate – von dynamischen Lua-Skripten für große HPC-Systeme über Compile-Time-Konfigurationen für Safety-enabled Systeme (ASIL-B) bis hin zu größenoptimierten tinit-Konfigurationen für MPU-Systeme (ASIL-D). Das Werkzeug ist gleichermaßen für die Konfiguration von virtuellen Maschinen, Containern und nativen Anwendungen geeignet.

Aufbauend auf dem Konfigurationswerkzeug wurden Designkonzepte für die Integration von Safety-Checks erarbeitet (Safety Configuration Checker, AP5.3.5/AP5.3.6). Hierbei lag der Fokus auf der visuellen Darstellung von Safetypartitionen und der automatisierten Prüfung gegen ein konfigurierbares Regelwerk, das die Anforderungen eines Safetymanuals abbilden kann. Eine vollständige prototypische Umsetzung der visuellen Safety-Hints war im verbleibenden Projektbudget nicht mehr realisierbar; die erarbeiteten Konzepte und Designentscheidungen bilden jedoch eine belastbare Grundlage für eine Weiterführung außerhalb des Projektkontexts, etwa in Folgeprojekten.

Test- und Safety-Infrastruktur

Die Test- und Safety-Infrastruktur von Kernkonzept wurde im Projektverlauf erheblich erweitert. Die Integration des Safety-Toolings (Requirements Tracing, Coverage Reporting) in den allgemeinen Testworkflow wurde verbessert und in Teilen als Open Source veröffentlicht (Kernel-Coverage-Werkzeuge und deren Integration). Die Infrastruktur wurde so erweitert, dass auch Userland-Komponenten Coverage-Informationen im Continuous Testing bereitstellen können. Das Tooling für das Anforderungstracing wurde um Mechanismen erweitert, welche die Safetyeigenschaften innerhalb des Systems korrekt zu den entsprechenden Requirements und deren Tests tracen können; auch hier wurde eine Integration in das Continuous Testing erreicht.

Die im Projekt geschaffene Test- und Benchmarkinfrastruktur ermöglicht die kontinuierliche Evaluation der Leistungsparameter des Systems im Rahmen der Continuous Integration und stellt damit sicher, dass Performance-Regressionen frühzeitig erkannt werden. Diese Infrastruktur kommt nicht nur den im Projekt entwickelten Mechanismen zugute, sondern wird auch über das Projektende hinaus für die Weiterentwicklung des L4Re-Systems genutzt.

Beiträge zur Anforderungs- und Architekturentwicklung

Im Rahmen der Arbeitspakete AP1, AP2 und AP3 hat Kernkonzept die Sicht des Hypervisors und Betriebssystems in die Anforderungserhebung und das Architekturdesign des Verbundprojekts eingebracht. Dies betraf zum einen Anforderungen an die Hardware aus Sicht der Virtualisierung und Isolation – etwa zur Sicherstellung von Freedom from Spatial Interference und Freedom from Temporal Interference auf High-Performance-Knoten – und zum anderen die Anforderungen an die Systembasissoftware seitens der Projektpartner.

Im dritten Projektjahr lag ein Schwerpunkt auf der Beratung der Projektpartner bei der Demonstratorentwicklung. Es wurden Empfehlungen zur sicheren Integration von Hypervisor und Betriebssystem in die jeweiligen Demonstratoraufbauten gegeben sowie Hinweise zur Auswahl geeigneter Hardware- und Softwarekomponenten. Speziell im Kontext von AP3.3 wurden potenzielle Seitenkanäle untersucht; die von den Projektpartnern betrachteten Seitenkanäle wurden auf ihre Anwendbarkeit auf den Hypervisor geprüft. Eine Härtung des Hypervisors selbst war auf Basis dieser Untersuchungen nicht erforderlich; die Erkenntnisse fließen jedoch in die Security Guidance für den sicheren Einsatz des Hypervisors ein.

Integration mit den Demonstratoren des Verbundprojekts

Die Integration der Kernkonzept-Beiträge mit den Ergebnissen der Projektpartner wurde im dritten Projektjahr umgesetzt. Die Beiträge von Kernkonzept konnten sowohl auf der x86_64- als auch auf der arm64-Architektur in verschiedenen Formaten demonstriert werden. Im gemeinsamen Demonstrator des Verbundprojekts konnten aufgrund der dort gewählten AMD-Hardware die Virtualisierungserweiterungen jedoch nicht gezeigt werden, da die Virtualisierungsmechanismen von AMD-Prozessoren (AMD-V/SVM) – im Gegensatz zu den entsprechenden Mechanismen von Intel-Prozessoren (Intel VT-x/VMX) – derzeit nicht von L4Re unterstützt werden.

Über die Integration der eigenen technischen Beiträge hinaus beteiligte sich Kernkonzept – ohne eigene budgetierte Aufwände in AP9 – an der inhaltlichen Ausgestaltung der Demonstratoren des Verbundprojekts. Dieser Beitrag erfolgte primär aus Story-Perspektive durch die Einbringung der Hypervisor- und Betriebssystemsicht in die Demonstratorerzählung.

2.4 Notwendigkeit und Angemessenheit der geleisteten Projektarbeiten

Die im Teilvorhaben geleisteten Arbeiten waren notwendig, um die in der Aufgabenstellung definierten Forschungs- und Entwicklungsziele zu erreichen und damit den Beitrag von Kernkonzept zu den Verbundzielen 4 (HW/SW-Codesign inklusive Betriebssystemintegration), 8 (ASIL-D auf Systemebene) und 9 (Situationsgerechter EE-Bordnetzbetrieb) sicherzustellen.

Die zu Projektbeginn formulierten Anforderungen an einen Hypervisor für Central-Car-Server-Architekturen – insbesondere die Sicherstellung von Freedom from Spatial und Temporal Interference unter Mixed-Criticality-Bedingungen, die effiziente Cross-Node-Kommunikation zwischen heterogenen Recheneinheiten sowie die nachweisbare Einhaltung von Safetyanforderungen durch geeignete Werkzeugunterstützung – konnten mit dem zu Projektbeginn vorhandenen Stand des L4Re-Systems nicht in vollem Umfang erfüllt werden. Die durchgeführten Entwicklungsarbeiten an den Schedulingmechanismen, am DRQ-Subsystem, an der Cross-Node-Kommunikation und am Konfigurationstooling waren daher unmittelbar erforderlich, um die Voraussetzungen für den Einsatz des Systems in zukünftigen Automotive-HPC-Plattformen zu schaffen.

Die Arbeiten zur Test- und Safety-Infrastruktur sind ebenfalls als notwendig einzustufen: Ohne eine kontinuierliche, automatisierte Evaluation der Performance- und Safetyeigenschaften können die im Projekt entwickelten Mechanismen weder zuverlässig weiterentwickelt noch im Hinblick auf Safetyanforderungen nachgewiesen werden. Die geschaffene Infrastruktur ist damit eine zwingende Voraussetzung sowohl für die Projektergebnisse selbst als auch für deren spätere Verwertung.

Die Beiträge zur Anforderungs- und Architekturentwicklung in den Arbeitspaketen AP1, AP2 und AP3 waren mit einem vergleichsweise geringen Personenmonatsaufwand (insgesamt 12 PM von 72 PM) angesetzt. Dieser Aufwand war erforderlich, um die Sicht des Hypervisors und Betriebssystems frühzeitig in die Hardware- und Architekturentscheidungen des Verbundprojekts einzubringen und damit spätere kostspielige Anpassungen oder Inkompatibilitäten zu vermeiden. Die enge Verzahnung zwischen Hardware- und Softwareentwurf ist für den Erfolg eines HW/SW-Codesign-Projekts wie CeCaS unerlässlich.

Die Verteilung der eingesetzten Personenmonate auf die Arbeitspakete entsprach der ursprünglichen Planung. Der Schwerpunkt lag mit 60 PM erwartungsgemäß auf AP5 (Softwarearchitektur), in dem die zentralen technischen

Entwicklungsarbeiten durchgeführt wurden. Verschiebungen des Budgets zwischen den Teilarbeitspaketen waren nicht erforderlich.

Insgesamt waren die geleisteten Projektarbeiten sowohl in Art als auch in Umfang notwendig und angemessen, um die definierten Ziele des Teilvorhabens zu erreichen. Die erzielten Ergebnisse rechtfertigen die eingesetzten Mittel und schaffen eine belastbare Grundlage für die im Verwertungsplan beschriebene weitere Nutzung der Projektergebnisse.

2.5 Voraussichtlicher Nutzen im Sinne des fortgeschriebenen Verwertungsplans

2.5.1 Erfindungen, Schutzrechtsanmeldungen und erteilte Schutzrechte

Kernkonzept hat im Rahmen des Vorhabens keine Schutzrechte und/oder Patente angemeldet oder erteilt bekommen.

2.5.2 Wirtschaftliche Erfolgsaussichten nach Projektende

Die im Teilvorhaben erzielten Ergebnisse schaffen die Voraussetzungen für den Einsatz des L4Re-Mikrokernsystems in zukünftigen Automotive-HPC-Plattformen. Erste lizenzierbare Produkte auf Basis der entwickelten Werkzeuge zur Konfiguration und Partitionierung des Hypervisors werden unmittelbar nach Projektabschluss erwartet. Zielgruppe sind insbesondere Hersteller, die Produkte auf Basis des L4Re-Hypervisors in Märkten mit Safety- und Securityanforderungen etablieren wollen. Folgeprodukte und Dienstleistungsangebote auf Basis der Open-Source-Erweiterungen werden zwei bis vier Jahre nach Projektabschluss erwartet.

Wettbewerbsvorteile ergeben sich aus der Kombination aus quelloffener Verfügbarkeit, den nachweisbaren Isolationseigenschaften mikrokern-basierter Systeme und den vorliegenden Sicherheits- und Safetyzertifizierungen des L4Re-Systems. Anwendergruppen sind insbesondere die deutsche und europäische Automobilindustrie sowie deren Tier-1-Zulieferer; eine Übertragbarkeit auf Bereiche wie Robotik, Medizintechnik und Industrie 4.0 ist gegeben.

2.5.3 Wissenschaftliche und technische Erfolgsaussichten nach Projektende

Die wissenschaftlichen Ergebnisse zu erweiterten Schedulingmechanismen werden in der Veröffentlichung “Scheduling Constraints: A Universal OS Mechanism for Managing Shared Resources” auf der RTAS 2026 der internationalen Forschungsgemeinschaft zugänglich gemacht. Die technischen Ergebnisse – insbesondere die Erweiterungen des L4Re-Mikrokerns – stehen über den Open-Source-Hauptentwicklungszweig akademischen wie industriellen

Stellen für eigene Forschungsarbeiten zur Verfügung. Die im Projekt geschaffene Test- und Safety-Infrastruktur unterstützt Folgeforschungen unmittelbar.

2.5.4 Wissenschaftliche und wirtschaftliche Anschlussfähigkeit

Wissenschaftlich besteht unmittelbarer Anschluss an Folgeforschungen im Bereich Mixed-Criticality-Scheduling, sicherer Systemkonfiguration und Cross-Node-Kommunikation. Die im Projekt erarbeiteten Konzepte zur visuellen Darstellung von Safetyeigenschaften im Konfigurationswerkzeug bilden eine Grundlage für Folgeprojekte, in denen diese Konzepte vollständig umgesetzt und in industrielle Werkzeugketten integriert werden können.

Wirtschaftlich umfassen die nächsten Schritte die Integration der entwickelten Werkzeuge in die Entwicklungsketten von OEMs und Tier-1-Zulieferern sowie die produktive Nutzung der erweiterten Schedulingmechanismen in konkreten Fahrzeugplattformen. Folgeprojekte zur Vertiefung dieser Schritte werden durch Kernkonzept aktiv verfolgt.

2.6 Dem Zuwendungsempfänger bekannt gewordener Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen

Im Berichtszeitraum sind dem Zuwendungsempfänger keine Forschungs- und Entwicklungsergebnisse von dritter Seite bekannt geworden, die für die Durchführung des Vorhabens unmittelbar relevant gewesen wären oder die im Projekt verfolgten Lösungsansätze in Frage gestellt hätten.

Die thematische Entwicklung im Bereich automotive-tauglicher Hochleistungsrechner und der dafür erforderlichen Systembasissoftware wird durch Kernkonzept im Rahmen seiner Beobachtung des Standes von Wissenschaft und Technik kontinuierlich verfolgt – insbesondere durch die Teilnahme an einschlägigen Konferenzen und durch den Austausch innerhalb der Safety- und Mikrokern-Community. Eine spezifische Beeinträchtigung der CeCaS-Projektarbeiten durch externe Entwicklungen lag jedoch nicht vor.

2.7 Erfolgte oder geplante Veröffentlichungen des Ergebnisses

Im Rahmen des Teilvorhabens wurden die folgenden Veröffentlichungen erzielt bzw. sind in Vorbereitung:

Wissenschaftliche Veröffentlichung:

Die im Rahmen der Arbeiten zu AP5.3.7 und AP5.3.8 erzielten Ergebnisse zu erweiterten Schedulingkontexten wurden in einer wissenschaftlichen

Veröffentlichung mit dem Titel “Scheduling Constraints: A Universal OS Mechanism for Managing Shared Resources” zusammengefasst. Die Veröffentlichung wird auf der IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS) 2026 erscheinen.

Open-Source-Veröffentlichungen:

Die im Projekt entwickelten Erweiterungen des L4Re-Mikrokernsystems werden im Rahmen der Open-Source-Strategie von Kernkonzept öffentlich zugänglich gemacht. Veröffentlichungen erfolgen in den Repositories der Kernkonzept-Organisation auf GitHub[1]. Ein Beispiel ist die Implementierung des tickless Idle-Modus, die im Hauptentwicklungszweig des L4Re-Mikrokerns Fiasco veröffentlicht wurde. Weitere Erweiterungen – etwa die Mechanismen zur Cross-Node-Kommunikation und die überarbeiteten Schedulingmechanismen – werden im Rahmen der regulären Releases des L4Re-Systems schrittweise als Open Source bereitgestellt.

[1] https://github.com/kernkonzept