VE-HEP

Härtung der Wertschöpfungskette durch quelloffene, vertrauenswürdige EDA-Tools und Prozessoren

FKZ: 16KIS1340

Kurzbericht per 31.08.2024

ZE:

Partnername: IAV GmbH

Vorhabenbezeichnung:

Härtung der Wertschöpfungskette durch quelloffene, vertrauenswürdige EDA-Tools und Prozessoren

Teilprojekt:

Demonstrator: Smart Sensor Communication

Laufzeit des Vorhabens:

01.03.2021 bis 31.08.2024

Impressum:

Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung gefördert.!

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren einfügen.

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

Inhaltsverzeichnis

Kurzbericht

Im Forschungsvorhaben HEP „Härtung der Wertschöpfungskette durch quelloffene, vertrauenswürdige EDA-Tools und Prozessoren“, Laufzeit 03/2021 bis 08/2024 übernahm IAV in erster Linie die Bearbeitung des Arbeitspakets 9 – „Demonstrator – Smart Sensor Communication“. Im Mittelpunkt des Projekts stand, mithilfe von Open-Source Mitteln und Methoden einen RISC-V Prozessor mit kryptografischer Peripherie zu entwerfen, welcher als Hardware Security Modul genutzt werden kann. Um den Nachweis der Nutzbarkeit zu erbringen, wurden im Arbeitspaket 9 Anforderungen an ein Testsystem und entsprechende Tests erhoben. Der Fokus der durchgeführten Arbeiten wurde im Verlauf des Projektes auf die Funktionalität des gefertigten Prozessors gelegt und konnte positiv nachgewiesen werden.

1 Aufgabenstellung

Die Aufgaben von IAV im HEP-Projekt konzentrierten sich, neben Zuarbeiten in den Arbeitspaketen 2 und 10, auf das Arbeitspaket 9 „Demonstrator – Smart Sensor Communication“ bei dem es das Ziel war, eine verschlüsselte Sensorkommunikation mithilfe des entwickelten Hardware Security Moduls zu demonstrieren sowie die Eigenschaften des Systems hinsichtlich Performance und weiteren Metriken zu überprüfen.

Das Arbeitspaket 9 wurde in sechs Unterarbeitspakete aufgeteilt, in denen jeweils der Hardware-Aufbau, die Software, die Inbetriebnahme, Testplanung sowie Durchführung und Dokumentation der jeweiligen Arbeiten beschrieben wurden. Die Unterarbeitspakete lauteten:

AP9.a: Aufbau und Inbetriebnahme des Demonstrators (FPGA Eval)
AP9.b: Entwicklung der Demonstrator Software
AP9.c: Planung der Testszenarien
AP9.d: Aufbau und Inbetriebnahme des Demonstrators (ASIC)
AP9.e: Durchführen der Tests und Benchmarks
AP9.f: Auswertung und Dokumentation

2 Ablauf des Vorhabens

Die Zuarbeiten im Arbeitspaket 2 konnten termingerecht abgeschlossen werden. Die Aufgaben für den Demonstratoraufbau (Arbeitspaket 9) wurden, wie geplant gestartet und konnten mit wenigen Verzögerungen im 2. Projektjahr durchgeführt werden. Durch diese Verzögerungen wurde der Durchführungszeitraum in Abstimmung mit allen Projektpartnern und dem BMBF um 6 Monate verlängert.

Die Unterarbeitspakete zur Anforderungserhebung an die Demonstrator-Hardware (Arbeitspaket 2) und die Testsoftware (Arbeitspaket 9.b1), die Planung des Demonstratoraufbaus (Arbeitspakete 9.a und 9.d) und der Testspezifikationen (Arbeitspaket 9.c) verliefen planmäßig. Durch rechtzeitige Beschaffung der Demonstrator Hardware konnte trotz Lieferengpässen und längeren Lieferzeiten der Aufbau gemäß der Planung durchgeführt werden. Während der Aufbauphase des Demonstrators entstand die Masterarbeit mit dem Titel „Validierungs- und Integrationsumgebung für Krypto-Beschleuniger“ in der die auszuführenden Arbeiten von Aygün Walter wissenschaftlich dokumentiert und validiert worden sind.

Im Zuge der Unterarbeitspakete zur Softwareerstellung und Testdurchführung (Arbeitspakete 9b und 9c) ist eine weitere Masterarbeit mit dem Titel „Sensorkommunikation eines Demonstratoraufbaus für ein neu entwickeltes Hardware Security Module auf Basis von RISC-V Standard“ durch Carl Schöps erstellt worden.

Die Bearbeitung weiterer Unterarbeitspakete (9.b2 bis 9.b6) führte zu einigen Verzögerungen. Ein problematischer Aspekt war die prototypische und unzureichend dokumentierte HSM-Firmware, welche die Erstellung der Schnittstelle zum AUTOSAR Modul erschwerte und verzögerte. Einige Funktionalitäten konnten deshalb erst nach erheblichem Testaufwand umgesetzt, oder mussten gänzlich gestrichen werden.

Verschiebungen ergaben sich auch bei der Bearbeitung der Testsoftware durch die Komplexität der genutzten AUTOSAR Umgebung. Die Erstellung und Konfiguration der notwendigen Module und deren Schnittstellen erforderte erheblich mehr Aufwand als geplant. Aufgrund der Tatsache, dass die Softwareentwicklung zu langsam war, wurde entschieden, eine unabhängige Architektur zu nutzen, um schneller die prototypischen Testfunktionalitäten am Demonstrator umsetzen zu können.

Weitere Verzögerungen der Bearbeitung wurden durch fehlerbehaftete Exemplare des ASICs verursacht. Fehlerhafte Speicher offenbarten unterschiedliches Verhalten zwischen FPGA und ASIC-Varianten. Bei bestimmten Fertigungen war die Kommunikationsschnittstelle komplett defekt. Hierbei nahm die Ursachenfindung auch einige Zeit in Anspruch. Nicht destotrotz konnten am Ende des Projektes mithilfe des Demonstratoraufbaus und der entwickelten Software kryptografische Funktionalitäten des gefertigten HEP-HSM nachgewiesen werden:

- Pseudo-Zufallszahlgenerierung
- Hash basierte Message Authentication Codes
- AES Ver- und Entschlüsselung mit vorkonfigurierten Schlüsseln.

3 Wesentliche Ergebnisse

Im Laufe des Projektes wurden durch die verschiedenen Arbeitspakete verschiedene Arbeitsergebnisse erstellt, diskutiert, dokumentiert und in den entsprechenden Verwaltungssystemen innerhalb des Konsortiums geteilt oder veröffentlicht. Im Arbeitspaket 2 „Anforderungserhebung“ wurden die relevanten Anforderungen an das zu testende System von allen Projektpartnern zusammengetragen und dokumentiert.

Weitere Anforderungsdokumente sind im Arbeitspaket 9 „Demonstrator: Smart Sensor Communication“ zusammengetragen worden. Hierbei sind in entsprechenden Unterarbeitspaketen die Anforderungen an das Demonstrator System, dessen Software sowie an die Testsoftware für das embedded System erstellt, diskutiert, geprüft und iterativ dem Projektstatus angepasst worden.

Ferner ist im Laufe des Projekts eine Vielzahl an Software-Lösungen (z.B. Skripte in automatisierten Umgebungen) entstanden, um den Status des Demonstratoraufbaus jederzeit überwachen bzw. abfragen zu können und um eventuelle Störquellen, falsche Konfigurationen oder bereits in Nutzung befindliche Komponenten zu überprüfen. Auch die Software für das embedded Device (Infineon Tricore Mikrocontroller) wurde so gestaltet, dass eine Wiederverwendbarkeit durch höchstmögliche Flexibilität der Softwarekonfiguration gewährleistet ist.

Im Arbeitspaket 10 entschieden die Projektpartner zur Veröffentlichung der Ergebnisse auf der Webseite des Onlinediensts GitHub ein Projekt zu erstellen: https://github.com/HEP-Alliance/. Hier werden die Ergebnisse veröffentlicht, um eine weitere Bearbeitung durch Projektpartner und Interessierte der Öffentlichkeit sicherzustellen. Die von IAV erstellte Software wird im eingerichteten Github Projekt (Arbeitspaket 10) unter Open Source Lizenzen veröffentlicht und zur weiteren Nutzung oder Erweiterung zur Verfügung gestellt. Außerdem soll diese Software im DI-SIGN-HEP Projekt (Förderprojekt des BMBF, Laufzeit 06/2024 bis 05/2027) auch von IAV genutzt und erweitert werden.

Die entwickelten Funktionen des HSMs konnten aufgrund der beschriebenen Umstände ausschließlich auf deren Funktionalität getestet werden. Performance Tests zur Laufzeitmessung der verschiedenen kryptografischen Funktionen wurden vorbereitet und konnten auch Ergebnisse liefern, wurden aber aus Zeitgründen, nicht vollumfänglich und reproduzierbar ausgeführt. Weiterführende Untersuchungen wie Angriffsszenarien und Tests für Manipulations- oder Anomalie-Erkennung (Arbeitspaket 9.c1) wurden, aufgrund der Priorisierung funktionaler Implementierungen, ebenfalls nicht umgesetzt.

Abschließend wurde der Funktionsnachweis eines neu erstellten und gefertigten Hardware Security Moduls mit Open-Source Methoden auf Basis von RISC-V Technologie erbracht. Darauf aufbauend können neue Technologien erarbeitet und entwickelt oder die bestehende Sicherheitsmethodik in neue Produkte integriert werden.

VE-HEP

Härtung der Wertschöpfungskette durch quelloffene, vertrauenswürdige EDA-Tools und Prozessoren

FKZ: 16KIS1340

Eingehende Darstellung per 31.08.2024

ZE:

Partnername: IAV GmbH

Vorhabenbezeichnung:

Härtung der Wertschöpfungskette durch quelloffene, vertrauenswürdige EDA-Tools und Prozessoren

Teilprojekt:

Demonstrator: Smart Sensor Communication

Laufzeit des Vorhabens:

01.03.2021 bis 31.08.2024

Impressum:

Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung gefördert.

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren einfügen.

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

Eingehende Darstellung

1 Ausführliche Darstellung der durchgeführten Arbeiten

Es werden neben den Zuarbeiten zu den Arbeitspaketen 2 und 10 die Unterkapitel der bearbeiteten Unterarbeitspakete des Arbeitspakets 9 dargestellt.

2 Zuarbeiten Arbeitspaket 2

Im Arbeitspaket 2 hat IAV gemeinsam mit den Partnern des Konsortiums ein Anforderungsdokument erstellt. Dabei wurden die Anforderungen in verschiedene Kategorien unterteilt und den jeweiligen Bearbeitenden der Arbeitspakete zugewiesen. Anwendungsfälle für den Demonstrator wurden entwickelt, aus denen weitere Anwendungsfälle und schließlich Anforderungen für das Hardware Security Modul sowie die Wertschöpfungskette abgeleitet wurden.

Die spezifischen Anforderungen an die für den Demonstrator relevante Hardware und deren Eigenschaften wurden insbesondere mit dem Partner Elektrobit ausführlich erarbeitet und dokumentiert. Zudem haben die Konsortialpartner Angriffsszenarien sowie mögliche Härtungsmethoden dokumentiert, die mithilfe des Demonstrators untersucht und bewertet werden können. Die Durchführung und Demonstration dieser Szenarien wird in den folgenden Arbeitspaketen erarbeitet und deren Realisierungsmöglichkeiten bewertet.

Eine Aufteilung der Systemanforderungen erfolgte für die folgenden Kategorien:

- Allgemeine (11 Anforderungen, 10 überprüft/akzeptiert, 1 verworfen)
- Application-Board (12 Anforderungen, 12 überprüft/akzeptiert)
- Debugger (0 Anforderungen – gekaufte Lösungen verwendet)
- FPGA-Board (3 Anforderungen, 3 überprüft/akzeptiert)
- Integrationsrechner (7 Anforderungen, 7 überprüft/akzeptiert)
- Mess- und Kontrollsystem (7 Anforderungen, 7 überprüft/akzeptiert)
- Sensor (9 Anforderungen, 9 überprüft/akzeptiert)

2.1 Unterarbeitspaket 9.a Aufbau und Inbetriebnahme des Demonstrators (FPGA)

Um die Anforderungen aus Arbeitspaket 2 (siehe Ausführungen im letzten Zwischenbericht) umzusetzen, wurde ein Systemaufbau aus verschiedenen Komponenten erarbeitet. Die verschiedenen Eigenschaften und Funktionen dieser Komponenten wurden intern und mit dem Projektpartner Elektrobit (zuständig für Arbeitspaket 8) diskutiert und festgehalten.

Ausgehend davon wurden für die einzelnen Komponenten Anforderungen erarbeitet. Anschließend wurde diejenige Hardware beschafft, die am ehesten den Anforderungen entspricht.

Abschließend wurden die Komponenten eingerichtet, konfiguriert und zu einem Demonstrator-Aufbau verbunden. Während der Durchführung der Anforderungsvalidierung kam es zu kleineren Anpassungen des Systemaufbaus sowie zu Erweiterungen und Ersetzungen von Hardwarelösungen im Bereich der Mess- und Kontroll-Lösungen sowie den verwendeten Schnittstellen. Diese Risiken wurden frühzeitig eingeplant und führten zu keiner relevanten Verzögerung oder Kostenänderung des Arbeitspakets**.**

Im weiteren Verlauf des Projekts wurde das Linux „Mess- und Kontroll-Systems“ von einem NVIDIA Jetson Nano System mit ARM-Architektur auf ein proprietäres x64 System umgezogen um RISC-V kompatible Entwicklungstools und andere Software, welche nicht für die ARM-Architektur zur Verfügung steht, schneller und einfacher nutzen zu können. Weiterhin wurden Anpassungen an Messmitteln und Schnittstellen vorgenommen, zum Beispiel der Austausch der Arduino Lösung zur Signalmanipulation durch Analog Discovery 2 Hardware.

Im Rahmen des Unterarbeitspakets wurde im Zeitraum von 11-2022 bis 05-2023 von Aygün Walter, Student an der Berliner Hochschule für Technik (BHT) und Masterand bei IAV, die Masterarbeit „Validierungs- und Integrationsumgebung für Krypto Beschleuniger“ erstellt. In der Arbeit werden die Anforderungen und Validierungsmethoden zum Demonstrator-Aufbau untersucht, beschrieben und ausgewertet.

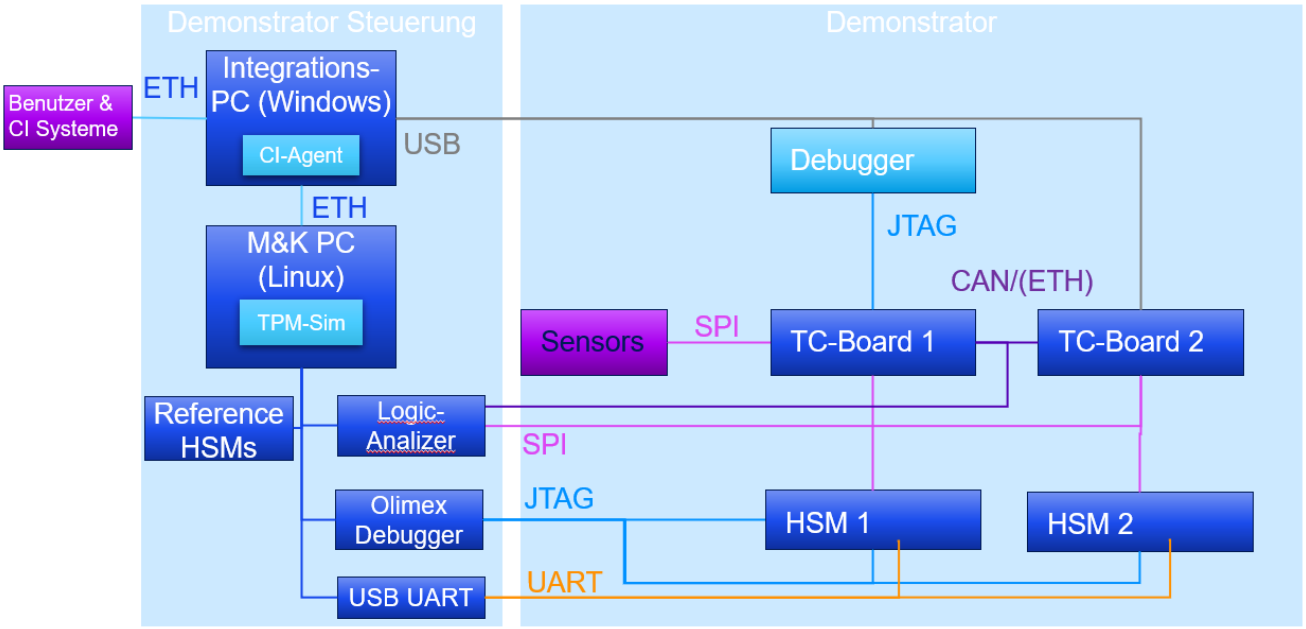

Abbildung 1: Übersicht des Demonstrator Hardware Aufbaus

2.2 Unterarbeitspaket 9.b Entwicklung der Demonstrator Software

Die vom Konsortium erarbeiteten Anforderungen aus AP2, sowie die Ergebnisse und Erkenntnisse aus dem Arbeitspaket 9a sowie Arbeitspaket 9c bilden die Grundlage für die Anforderungen an die Demonstrator Software. In diesem Arbeitspaket gab es die meisten unvorhergesehenen Verzögerungen oder notwendigen Änderungen. Dies lag unter anderem an mangelnden Zuarbeiten bezüglich der HSM-Firmware und der unvorhergesehenen hohen Komplexität von Architekturplatform-Software, welche in Zusammenarbeit mit dem Partner Elektrobit verwendet wurde. Hinzukommend nahm die Fehlersuche beim Inbetriebnehmen der gefertigten HSM ASICs viel Zeit in Anspruch, da es verschiedene Phänomene und Fehlerfälle gab, welche ausführlich untersucht werden mussten. Auch einige Anpassungen im Demonstratoraufbau mussten aufgrund ungeeigneter Komponenten oder Schnittstellen vorgenommen werden, welche die Softwareentwicklung verzögerten.

Im Rahmen des Unterarbeitspakets 9 ist die Masterarbeit von Carl Schöps „Sensorkommunikation eines Demonstratoraufbaus für ein neu entwickeltes Hardware Security Module auf Basis des RISCV Standards“ entstanden, bei dem ein Teil der Tests implementiert wurde und anhand eines Referenz-HSMs evaluiert worden ist.

2.2.1 Unterarbeitspaket 9.b1 Anforderungen

Die Software-Anforderungen wurden initial erstellt, überarbeitet und Qualitätssicherungsmaßnahmen unterzogen.

Entsprechend der bereits erfolgten Arbeiten wurden die Software-Anforderungen für drei wesentlichen Domänen aufgeteilt. Durch die Komplexität der Applikationssoftware ist eine weitere Aufteilung der Anforderungen erfolgt.

1. Demonstrator Applikationssoftware
    a. Allgemeine Qualität
    b. Schnittstellen
    c. Unterstützte HSM-Funktionalitäten
    d. Monitoring

e. Sensor
f. Kommunikation

2. Demonstrator Integrationsumgebungssoftware
3. Demonstrator Mess- und Kontrollsystem Software

Für die Applikationssoftware wurde der Ansatz, zwei verschiedene Software-Varianten (für Sender mit Sensor / Empfänger ohne Sensor) zu erstellen verworfen und stattdessen der Ansatz einer universellen Variante verfolgt. Da die Mess- und Kontroll-Einheit nur einer Instanz Kommandos sendet, kann dadurch die Unterscheidung der Aufgaben besser gesteuert werden.

Die Anforderungen sind mit mehreren Eigenschaften wie Identifikationsnummer, Status und weiteren Eigenschaften in einem IAV internen Atlassian Confluence Dokumentationsbereich nachverfolgbar dokumentiert und überprüft. Sie bilden die Grundlage für die weitere Entwicklung von Validationstests, welche ebenfalls auf dem Confluence-Bereich mithilfe des Yogi Plugins angelegt wurden. Überarbeitungen und Änderung der Anforderungen und Validationstests werden, sofern notwendig, stetig aktualisiert. Die Anforderungen wurden den Projektpartnern präsentiert, diskutiert und wurden in wiederkehrenden Meetings abgestimmt.

2.2.2 Unterarbeitspaket 9.b2 Architektur

Eine grundlegende Architektur der Demonstrator Software wurde erarbeitet. Ebenso wurden grundlegende Coding- und Namenskonventionen diskutiert und entwickelt, die sowohl für die Software zur Steuerung und die Applikationssoftware gelten.

Bei der Applikationssoftware wurde die Abgrenzung von verschiedenen Software-Integrationen (für Smart-Sensor- und Steuerungs-Modul) als nicht-notwendig identifiziert. Die Priorität wurde auf hohe Anpassungsfähigkeit gelegt ohne verschiedene Versionen pflegen zu müssen. Während der Implementierung hat sich allerdings herausgestellt, dass die maßgeblich vorgegebene Architektur durch den AUTOSAR Standard und die zur Verfügung stehende Konfigurationssoftware inhärent komplex ist und eine dynamische Anpassung oder Änderungen nicht ohne erhebliche Aufwände erlaubt. Dementsprechend wurde ein neuer Ansatz parallel zur bisher erarbeiteten Lösung aufgesetzt und erarbeitet. Ein Überblick über die in drei Schichten erstellte Architektur und deren Module und Schnittstellen ist in Abbildung 2 dargestellt. Der Ansatz die AUTOSAR Seriensoftware für den Einsatz in dieser prototypischen Umgebung zu nutzen, wurde im Laufe des Projekts verworfen und durch diese dynamischere Lösung ersetzt.

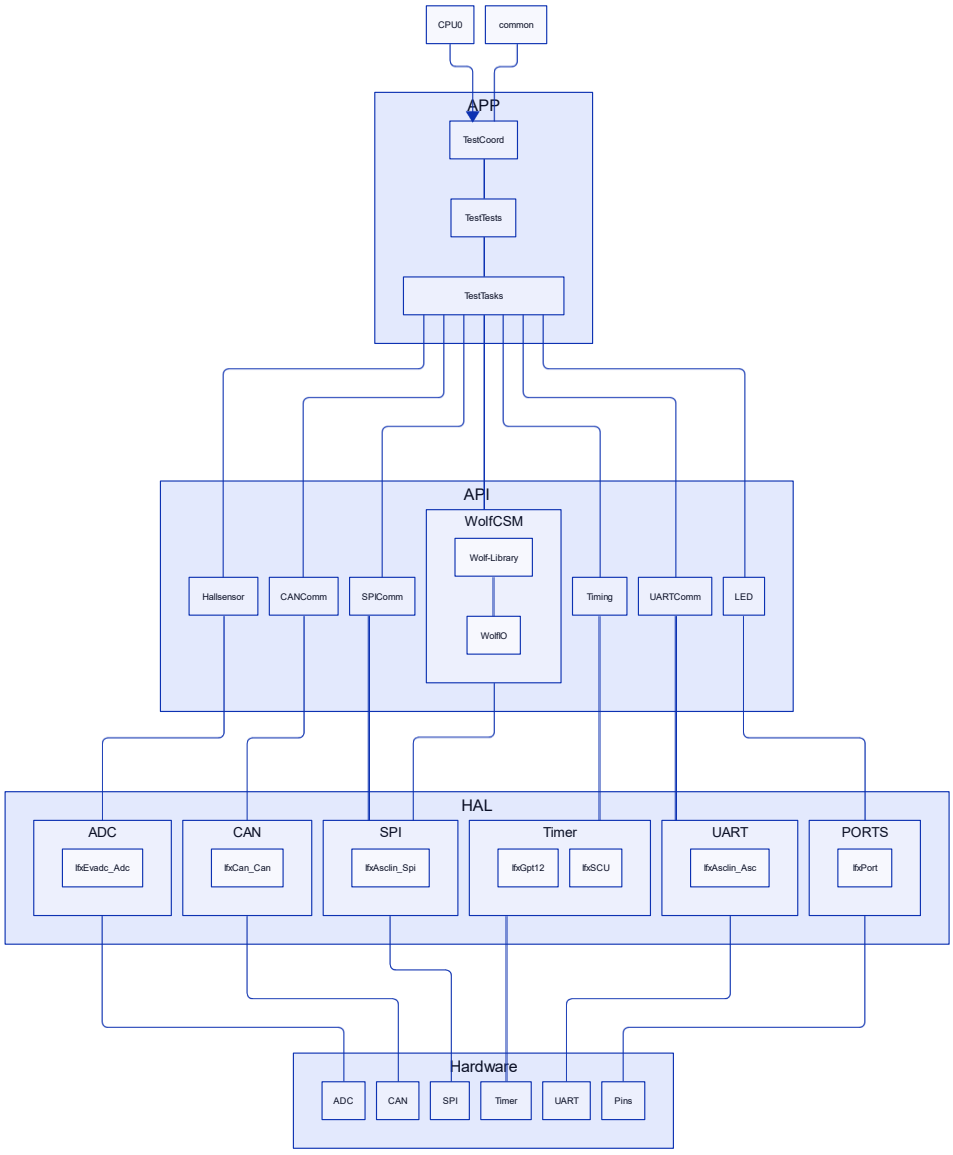

Abbildung 2: entwickelte Softwarearchitektur für das Applikations-Board (Infineon Tricore)

2.2.3 Unterarbeitspaket 9.b3 Implementierung

Für den Integrationsrechner und das Mess- und Kontrollsystem wurden Python-Skripte entwickelt, welche mithilfe einer eingerichteten Gitlab-Pipeline periodisch ausgeführt werden konnten, um den Status der verwendeten Hard- und Software überprüfen zu können. Hierzu wurden Jobs implementiert welche alle angeschlossenen Komponenten nach Verfügbarkeit, sowie exklusive Softwareinstanzen überprüfen und den Status in ein Visualisierungstool melden.

Die Implementierung der Applikationssoftwarekomponenten wurde im Projektjahr 2023 begonnen. Aufgrund des iterativen und agilen Ansatzes wurden die Umsetzungen stetig angepasst und optimiert sowie ergänzende oder neue Funktionalitäten hinzugefügt, um die Anforderungen bestmöglich umzusetzen. Hierbei sind auch die zeitintensiven Aufwände für Softwareänderungen der Applikationssoftware aufgefallen und es wurde sich entschieden die Architektur entsprechend neu aufzusetzen. Die nachfolgende Implementierung der Tests konnte mit geringerem zeitlichem Aufwand umgesetzt werden.

2.2.4 Unterarbeitspaket 9.b4 Test der Softwarekomponenten

Der Test der Softwarekomponenten wurde parallel zur Implementierung durchgeführt. Da zu den Anforderungen an die Applikationssoftware im Unterarbeitspaket 9.b.1 auch Verifikationskriterien spezifiziert wurden, konnte die Software kontinuierlich gegen diese getestet wurden.

Da die Applikationssoftware selbst zu Testzwecken genutzt wurde, war eine Differenzierung zwischen den Tests der Applikationssoftware und der Applikationssoftware welche Tests für den HSM umsetzt, essenziell um Probleme effizient diskutieren zu können. Die Tests zur Applikationssoftware wurden nicht in einer automatischen Testumgebung realisiert, sondern wurden lediglich manuell anhand der beschriebenen Testspezifikation durchgeführt. Ein Testfall ist beispielhaft in Abbildung 3 dargestellt.

Testfall	Status	Description	Description(English)	Requirement	Notes (Link to Bitbucket/Code/Log?)
SW-TEST-020	REVIEWED	**Zweck:** Überprüfen, ob die Software in der Lage ist, eine erfolgreiche Kommunikation mit dem HSM (Hardware Security Module) herzustellen. **Vorbedingungen:** • Das HSM ist ordnungsgemäß an das System angeschlossen. • Die Software ist auf dem entsprechenden System installiert und konfiguriert. **Testschritte:** 1. Starten Sie die Software und öffnen Sie das Modul zur Kommunikation mit dem HSM. 2. Konfigurieren Sie das Modul so, dass es eine Verbindung mit dem HSM herstellt. 3. Senden Sie Testbefehle an das HSM über die Software. 4. Überprüfen Sie, ob die Software die vom HSM zurückgegebenen Antworten korrekt verarbeitet hat. **Erwartungswerte:** • Die Software ist in der Lage, eine erfolgreiche Verbindung mit dem HSM herzustellen. • Die Software verarbeitet die vom HSM zurückgegebenen Antworten korrekt. **Testumgebung:** • Betriebssystem mit Softwarestack • Applikations-Software • Hardware Security Module • Kommunikationsschnittstelle **Testdaten:** • Testbefehle, die an das HSM über die Software gesendet werden sollen. **Testergebnisse:** • Erfolg: Die Software stellt eine erfolgreiche Verbindung mit dem HSM her und verarbeitet die vom HSM zurückgegebenen Antworten korrekt. • Fehler: Die Software kann keine Verbindung mit dem HSM herstellen oder verarbeitet die vom HSM zurückgegebenen Antworten nicht korrekt. • Ausnahme: Die Software stürzt während des Tests ab.	**Purpose:** Check whether the software is able to establish successful communication with the HSM (Hardware Security Module). **Prerequisites:** • The HSM is properly connected to the system. • The software is loaded and configured on the corresponding system. **Test steps:** 1. Configure the software to connect to the HSM. 2. Send test commands to the HSM through the software and wait for the response. 3. Verify that the software correctly processed the responses returned by the HSM. **Expected values:** • The software is able to establish a successful connection with the HSM. • It is possible to send test commands to the HSM through the software and receive the response after a certain wait time. • The software correctly processes the responses returned by the HSM. **Test environment:** • Operating system with software stack (EB AUTOSAR) • Application software • Hardware security modules • Communication interface **Test data:** Test commands to be sent to the HSM via the software. **Test results:** • Success: The software successfully connects to the HSM and correctly processes the responses returned by the HSM. • Error: The software cannot connect to the HSM or does not correctly process the responses returned by the HSM. • Exception: The software crashes during testing.	AP-SW-REQ-008 HEP-146 - SW-TEST-048: Checking the communication between Application Board - ASIC HSM APPROVED	• Kommunikationsschnittstelle bereiets Festgelegt: **SPI**

Abbildung 3: Beispiel Testfall Applikationssoftware

2.2.5 Unterarbeitspaket 9.b5 Integration der Softwarekomponenten und Softwaretest

Da die Software zur Überprüfung der Integrationsrechner und des Mess- und Kontrollsystems über eine Gitlab-Pipeline realisiert wurde, ist im Folgenden mit Integration lediglich die Erstellung von Maschinencode für die Applikationssoftware zu verstehen.

Die Applikationssoftware basierend auf der AUTOSAR Architektur wurde mithilfe von EB Tresos konfiguriert und integriert. Die Zusammenarbeit mit EB erfolgte über die Kundenanbindung in einer webbasierten Plattform. Hierbei wurde von EB eine neue Softwarevariante ausgeliefert, sobald es Änderungen an den von EB entwickelten Softwarekomponenten gab, oder sobald wir Ihnen Änderungen an unseren Softwarekomponenten gespiegelt haben.

Somit konnten wir unsere Softwarekomponenten nur mit einer gewissen Verzögerung nach der erfolgten Konfiguration integrieren und testen. Um eine schnellere Iteration unserer Arbeitsergebnisse zu erzielen, wurde parallel an einer unabhängigen Umsetzung mit Infineon Aurix-Studio IDE entwickelt. Die Entwicklung einer Architektur und der zugehörigen Komponenten erfolgte anhand von Best Practices und Erfahrungen mit früheren Projekten und konnte schnell durchgeführt werden. In dieser Umgebung konnten auch mehrere Entwickler effektiv parallel arbeiten, da die Software in einem internen Gitlab-Repository abgelegt wurde.

2.2.6 Unterarbeitspaket 9.b6 Systemintegration und Integrationstest

Die Systemintegration wurde parallel bei EB und IAV durchgeführt. Wobei der Integrationstest lediglich die Durchführung der Applikationssoftware darstellt, da der Funktionsnachweis des HSM ein wesentlicher Teil der Demonstration ist.

Eine Steuergeräte-Instanz wurde in den Demonstrator-Aufbau integriert und die zugehörigen Integrationstests vorbereitet, implementiert und durchgeführt. Die Verbindung zu einer zweiten Instanz ist, aufgrund niedriger Priorität, nicht erfolgt, somit sind die Integrationstests für den „Fahrzeug-Bus“ lediglich mithilfe des Mess- und Kontroll-Systems getestet.

2.3 Unterarbeitspaket 9.c Planung der Testszenarien

Zur Planung der Testszenarien wurde ein Testmanagementplan erstellt, der die Grundlage für alle Aufgaben im Umfeld des Testens bildet. In diesem sind die Testszenarien in die zwei Testprozesse:

- **a)** Robustheit und Manipulation (Unterarbeitspaket AP9c.1)
- **b)** Performanz (Unterarbeitspaket AP9c.2),

sowie in drei Teststufen:

- **1.)** HSM-Applikation,
- **2.)** Steuergeräte-Applikation
- **3.)** Kommunikation

unterteilt.

Primär wurden die Performanz Tests für die HSM-Applikation spezifiziert, welche die Funktionalität des jeweiligen Testgegenstands voraussetzen. Für die insgesamt 18 Tests für die beiden Metriken Zeit- und Stromverbrauch wurde der Testablauf beschrieben. Hierbei wurde sich an gängige Praxis gehalten und jeder Test in den Abschnitten: Beschreibung, Zweck, Vorbedingungen, Testschritte, Erwartete Ergebnisse, Testumgebung, Testdaten und Testergebnisse ausführlich beschrieben. Die entwickelten Tests wurden in Reviews diskutiert, wenn nötig angepasst und auf einer Confluence Seite mit Historie abgelegt.

2.4 Unterarbeitspaket 9.d Aufbau und Inbetriebnahme des Demonstrators (ASIC)

Der grundlegende Aufbau des Demonstrators wurde so aufgebaut, dass für die Verwendung des gefertigten ASICs lediglich die Hardwareschnittstelle des FPGA auszutauschen ist. Diese Lösung ermöglicht ein schnelles Austauschen der HSM-Instanz und einen Vergleich.

Im Laufe des Projekts wurden einige Komponenten angepasst, ausgetauscht oder konnten zusammengefasst werden. So wurden beispielsweise die Überwachungen der seriellen, SPI und CAN-Schnittstellen alle mithilfe des Analog Discovery Messmittels realisiert werden, daraus folgt eine Vereinfachung des Aufbaus und der zu entwickelnden Softwaretests.

Im Rahmen des Aufbaus der Komponenten wurden Tests mithilfe der in AP 9b5 entwickelten CI/CD Lösung entwickelt, um den Demonstrator Aufbau automatisiert zu überprüfen und Fehler frühzeitig zu erkennen und zu visualisieren. Die verschiedenen Tests werden automatisiert getriggert, sodass jederzeit erkennbar ist, ob Hardware-Komponenten oder Software-Ressourcen korrekt angeschlossen, konfiguriert und verfügbar sind. Ein beispielhaftes Ergebnis eines Pipeline Durchlaufs zum Selbsttest ist in Abbildung 4 dargestellt, die Pipeline ist in die vier Stufen: Precheck, Connection, Availability und Deploy aufgeteilt. Hier wird erst die Verbindung zum angeschlossenen Linux Mess- und Kontrollrechner überprüft, anschließend werden die angeschlossenen Komponenten wie Messhardware, Debugger oder andere Schnittstellen überprüft. Nachfolgend wird bei exklusiv verwendbaren Komponenten deren aktuelle Verfügbarkeit getestet. Abschließend werden die Testergebnisse zur Visualisierung in eine Datenbank geschrieben.

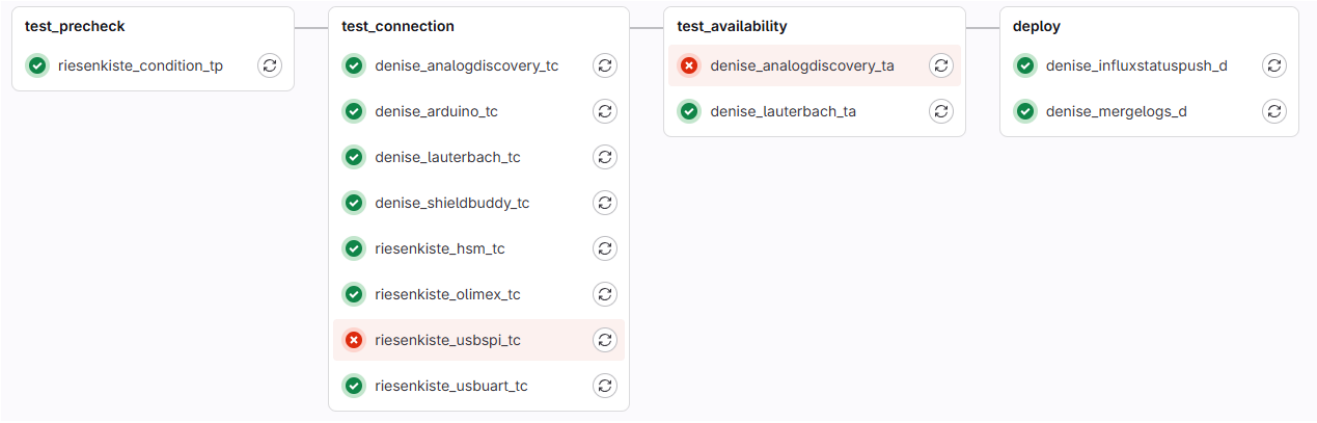

Abbildung 4: gitlab Pipeline zum Selbsttest der Hard- und Software

2.5 Unterarbeitspaket 9.e Durchführen der Tests und Benchmarks

Bei der Durchführung der Tests wurden die funktionalen Aspekte der HSM-Fähigkeiten gegenüber den Performance Metriken priorisiert. So wurden zwar testweise Laufzeiten für die Generierung von Zufallszahlen oder Hashwerten gemessen, aber nicht ausgiebig und verifizierbar genug durchgeführt, stattdessen wurde sich um die Überprüfung der weiteren kryptografischen Funktionalitäten konzentriert.

Die Durchführung der Ver- und Entschlüsselung mithilfe der AES-Hardware war besonders zeitintensiv da die Einrichtung, die Kommandos, die zu verwendenden Schlüssel, deren Generierung und Ablageorte dieser sowie der Klartext und des Schlüsseltexts nicht, bzw. nicht zufriedenstellend, dokumentiert war. Die Durchführung der AES Ver- und Entschlüsselung konnte am Ende erfolgreich durchgeführt worden, allerdings sind durch den erheblichen zeitlichen Aufwand die restlichen Testumfänge wie der Stromverbrauch sowie die Robustheits- und Manipulationstests nicht durchgeführt werden.

2.6 Unterarbeitspaket 9.f Auswertung und Dokumentation

Die Dokumentation des Arbeitspakets 9 erfolgte hauptsächlich in einem eigenen Confluence Bereich, und in den jeweiligen Repositories der erstellten Software. Die Arbeiten umfassten insbesondere die Erstellung, und Aktualisierung aller Änderungen der Softwareentwicklungen in den entsprechenden Repositories sowie die strukturierte Dokumentation der Arbeitspakete in Confluence.

Zunächst wurden alle Softwareentwicklungen im Projekt in Bitbucket-Repositories abgelegt. Im Jahr 2023 erfolgte ein Umzug der Repositories nach GitLab. Dieser Umzug wurde durchgeführt, um mithilfe von Pipelines Automatisierungen umzusetzen, was eine flexiblere und bessere firmeninterne Unterstützung als die Kombination von Bitbucket und Bamboo ermöglichte. Jedes Repository enthält eine README-Datei im Markdown-Format, in der die Funktion, der Aufbau und die Verwendung der jeweiligen Software beschrieben sind. Zudem sind in den README-Dateien Verlinkungen zu entsprechenden Confluence-Seiten enthalten, die weiterführende Informationen bereitstellen.

Begleitende Dokumente zu allen Arbeiten wurden in einem separaten Repository namens „docs_sheets“ abgelegt und an den entsprechenden Stellen verlinkt. Die Arbeitspakete sind in Confluence strukturiert und dokumentiert, was eine übersichtliche und nachvollziehbare Darstellung der Projektfortschritte und -ergebnisse gewährleistet.

2.7 Zuarbeiten Arbeitspaket 10a

Um die Verbreitung der erzielten Ergebnisse in der Industrie sicherzustellen, wurde von den Projektpartnern ein Industrial Liaison Board eingerichtet. Dieses Board wurde initial mit den Industriepartnern des Konsortiums und den assoziierten Partnern besetzt und konnte während der Projektdauer mit anderen Industriepartnern erweitert werden. In regelmäßigen Online-Treffen wurden Mittel und Möglichkeiten diskutiert und erarbeitet, um die

Arbeitsergebnisse des Forschungsprojekts wirksam einem Fachpublikum sowie der Öffentlichkeit zu präsentieren. Eine geeignete Form der Veröffentlichung wurde in der Erstellung einer Webseite durch das Konsortium gefunden, auf der wesentliche Inhalte der Forschungspartner veröffentlicht werden. Zudem wurden Teilvorhaben und Ergebnisse des Vorhabens in Workshops vorgestellt und diskutiert.

Es fand jährlich mindestens ein physisches Statusmeetings statt. Wesentliche Ergebnisse der Forschungspartner wurden auf der Webseite http://hep-alliance.org veröffentlicht. Es wurde außerdem festgelegt, wie erstellte Arbeitsergebnisse veröffentlicht werden können. Hierzu wurde ein GitHub Projekt erstellt (https://github.com/HEP-Alliance/). Zudem wurde gemeinschaftlich an einer wissenschaftlichen Veröffentlichung in Form eines Papers für den IEEE-Workshop „Cyber Resilience and Economics (CRE)“ (https://www.ieee-csr.org/cre/) gearbeitet. Im November 2023 fand ein Konsortialtreffen in Berlin bei IAV statt. Hier wurde unter anderem die Erstellung eines wissenschaftlichen Papers im Rahmen der DATE-Konferenz 2024 (25. bis 27.03.2024) diskutiert und konzeptioniert. Der Titel des Papers lautete: „Evaluating an Open-Source Hardware Approach from HDL to GDS for a Security Chip Design – A review of the final stage of Project HEP” (https://ieeexplore.ieee.org/document/10546500).

Die Ergebnisse und Meilensteine des Projekts wurden kontinuierlich per Pressemitteilungen, Vorträgen oder auf Workshops vorgestellt und mit dem Fachpublikum diskutiert. Die Resonanzen waren stets positiv, so wurden diese zum Beispiel in Artikeln auf heise-online (https://www.heise.de/news/Open-Source-Hardware-Sicherheitschip-mit-offenen-Tools-in-Brandenburg-gefertigt-9340167.html), idw-online und anderen Medien aufgenommen und veröffentlicht. Wesentliche Ergebnisse des Forschungsvorhabens wurden ebenfalls auf der Webseite http://hep-alliance.org veröffentlicht.

3 Wichtige Positionen zahlenmäßiger Nachweis

Der Schlussverwendungsnachweis für das Projekt wurde Anfang des Jahres 2025 gestellt. Dieser zeigt, dass alle Kostenkategorien weitgehend eingehalten wurden. Es ergab sich jedoch eine kleinere Gesamtsumme für das Vorhaben, so dass Bundesmittel in Höhe von rund 12.000 EUR nicht abgerufen wurden. Dies liegt in der Tatsache begründet, dass Aufgabenschwerpunkte - wie weiter oben beschrieben - im Laufe des Projektes angepasst wurden.

4 Notwendigkeit der geleisteten Projektarbeiten

Der Verlauf der Arbeiten im Projekt HEP folgte weitgehend der im Antrag formulierten Planung. Notwendige Anpassungen wurden vorgenommen, um auf unvorhergesehene Herausforderungen zu reagieren. Die erbrachten Leistungen waren von entscheidender Bedeutung für die Realisierung der anspruchsvollen technischen Ziele des Projekts und wurden innerhalb der vorgegebenen bzw. aktualisierten Zeitpläne mit den vorgegebenen Ressourcen erbracht. Die im Arbeitsplan definierten Aufgaben wurden ohne zusätzliche Ressourcen erfolgreich bearbeitet.

Die enge Zusammenarbeit zwischen den Projektpartnern garantierte eine praxisnahe und fachlich fundierte Durchführung der Arbeiten. Die Partner brachten ihre spezifischen Expertisen ein, wodurch wesentliche Aspekte einer quelloffene Wertschöpfungskette erarbeitet und gehärtet worden sind. Durch Auswahl und Erweiterung bestehender EDA-Werkzeuge konnten Prozessoren gefertigt werden. Diese Kooperation war maßgeblich für den Projekterfolg und bestätigt die Angemessenheit der gewählten Projektstruktur.

5 Voraussichtlicher Nutzen

Durch den Open-Source Ansatz können die in dem Forschungsvorhaben erreichten Ergebnisse allen interessierten Entitäten zur Verfügung gestellt werden. Neben den Möglichkeiten, die sich vor allem in den Bereichen Wissenschaft und Forschung hervorheben lassen, können durch die geschaffene Alternative vor allem in der Industrie Abhängigkeiten von Zulieferern, bestimmten Teilen der Wertschöpfungskette oder sogar Nationalitäten abgebaut werden. Gleichzeitig wird durch die offene Entwicklung die Möglichkeit geschaffen, dass mögliche Sicherheitsrisiken oder Designfehler von Dritten gemeldet und mitigiert werden können.

Für IAV ist der Wissensaufbau in Zusammenhang mit RISC-V Hardware ein wichtiger Teil, um diese Technologien in zukünftigen Projekten anwenden zu können. Wir erhoffen uns durch die Verwendung von Open Source Hardware effizientere und sichere Lösungen schneller durch vorhandene und bereits getestete Teillösungen anbieten zu können. Deshalb haben wir uns auch entschieden, an einem weiteren Forschungsvorhaben, DI-Sign-HEP, mit vielen Konsortialpartnern des VE-HEP Projekts teilzunehmen, das die Themen des abgeschlossenen Projekts aufgreift und weitere Potentiale erforscht.

6 FuE-Ergebnisse bei Dritten

Open-Source Hardware

1. **Projekt: RISC-V Prozessoren**
    - **Autoren/Institutionen**: Verschiedene Universitäten und Unternehmen, darunter die University of California, Berkeley.
    - **Ziele**: Entwicklung einer offenen und lizenzfreien Prozessorarchitektur.
    - **Ergebnisse**: RISC-V hat sich als eine der führenden offenen Hardware-Architekturen etabliert, die in verschiedenen Anwendungen von IoT-Geräten bis hin zu Supercomputern eingesetzt wird.
2. **Projekt: OpenTitan**
    - **Autoren/Institutionen:** lowRISC CIC
    - **Ziele**: OpenTitan bietet eine Umgebung zur Produktion von Hard- und Software Designs welche fähig sind verschiedene Anwendungen zu realisieren. Hierzu gehören zum Beispiel ein sicherer Mikrocontroller als auch eine sichere integrierte Anwendungsumgebung mit Sicherheitsfunktionen (RoT, Secure Boot, etc.)
    - **Ergebnisse:** Hard- und Software Repositorien mit funktionalen Architekturen

3. **Projekt: Open Compute Project (OCP)**
    - **Autoren/Institutionen**: Facebook (jetzt Meta), über 300 andere große Technologieunternehmen (ARM, IBM, Intel, Google, Microsoft, NVIDIA uvm.)
    - **Ziele**: Entwicklung und Förderung von offenen Standards für Rechenzentrums-Hard- und Firmware.
    - **Ergebnisse**: Veröffentlichung mehrerer Hard- und Firmware-Designs und Spezifikationen, die von vielen Unternehmen weltweit übernommen wurden.
4. **Projekt: Caliptra (als Teil des CHIPS Alliance Projekts)**
    - **Autoren/Institutionen:** CHIPS Alliance (AMD, Google, Infineon, intel, Microsoft uvm.)
    - **Ziele:** IP und Firmware für einen integrierten „Root-of Trust“ Block
    - **Ergebnisse:** Aus OCP hervorgegangenes Projekt
        - Revisionen werden bei OCP publiziert
        - Arbeitsergebnisse werden auf GitHub gehostet

Open Source Cybersecurity

1. **Projekt: OpenSSL 3.0 (https://openssl-library.org/)**
    - **Autoren/Institutionen**: OpenSSL Project Team.
    - **Ziele**: Verbesserung der Sicherheit und Funktionalität der OpenSSL-Bibliothek.
    - **Ergebnisse**: Einführung neuer Sicherheitsfunktionen, verbesserte API und bessere Unterstützung für moderne kryptografische Algorithmen.
2. **Projekt: Zeek (ehemals Bro)**
    - **Autoren/Institutionen**: International Computer Science Institute (ICSI) und andere.
    - **Ziele**: Entwicklung eines leistungsstarken Netzwerküberwachungssystems.
    - **Ergebnisse**: Weit verbreitete Nutzung in der Netzwerküberwachung und -analyse, mit neuen Funktionen zur Erkennung und Abwehr von Cyberangriffen.
3. **Projekt: Let's Encrypt**
    - **Autoren/Institutionen**: Internet Security Research Group (ISRG).
    - **Ziele**: Bereitstellung kostenloser SSL/TLS-Zertifikate zur Förderung der Verschlüsselung im Internet.
    - **Ergebnisse**: Dramatischer Anstieg der verschlüsselten Websites, Verbesserung der allgemeinen Internetsicherheit.

Während des Projekts wurden insbesondere die Forschungsvorhaben und Ergebnisse der beiden Projekte OpenTitan und Caliptra der CHIPS Alliance verfolgt. Aufgrund der Erfahrungen im Bereich der Firmware wurde der Ansatz gewählt, eine der beiden Lösungen für ein mögliches Folgeprojekt nutzen zu können.

7 Veröffentlichungen

- IAV Interne Nachrichten im Intranet
- IAV LinkedIn Beiträge im Laufe des Projektes
- Beiträge auf http://hep-alliance.org

- Teilnahme am Tag der vertrauenswürdigen Elektronik 4. und 5.6.2024 (https://www.velektronik.de/tage-der-vertrauenswurdigen-elektronik-2024/)
- Teilnahme an Auftaktveranstaltung Chipdesign Germany in Hannover am 5.6.2024
- „Evaluating an Open-Source Hardware Approach from HDL to GDS for a Security Chip Design – A review of the final stage of Project HEP" (https://ieeexplore.ieee.org/document/10546500).