

# *The HERA Methodology: Coarse Grained Reconfigurable Logic for General Purpose Computing*

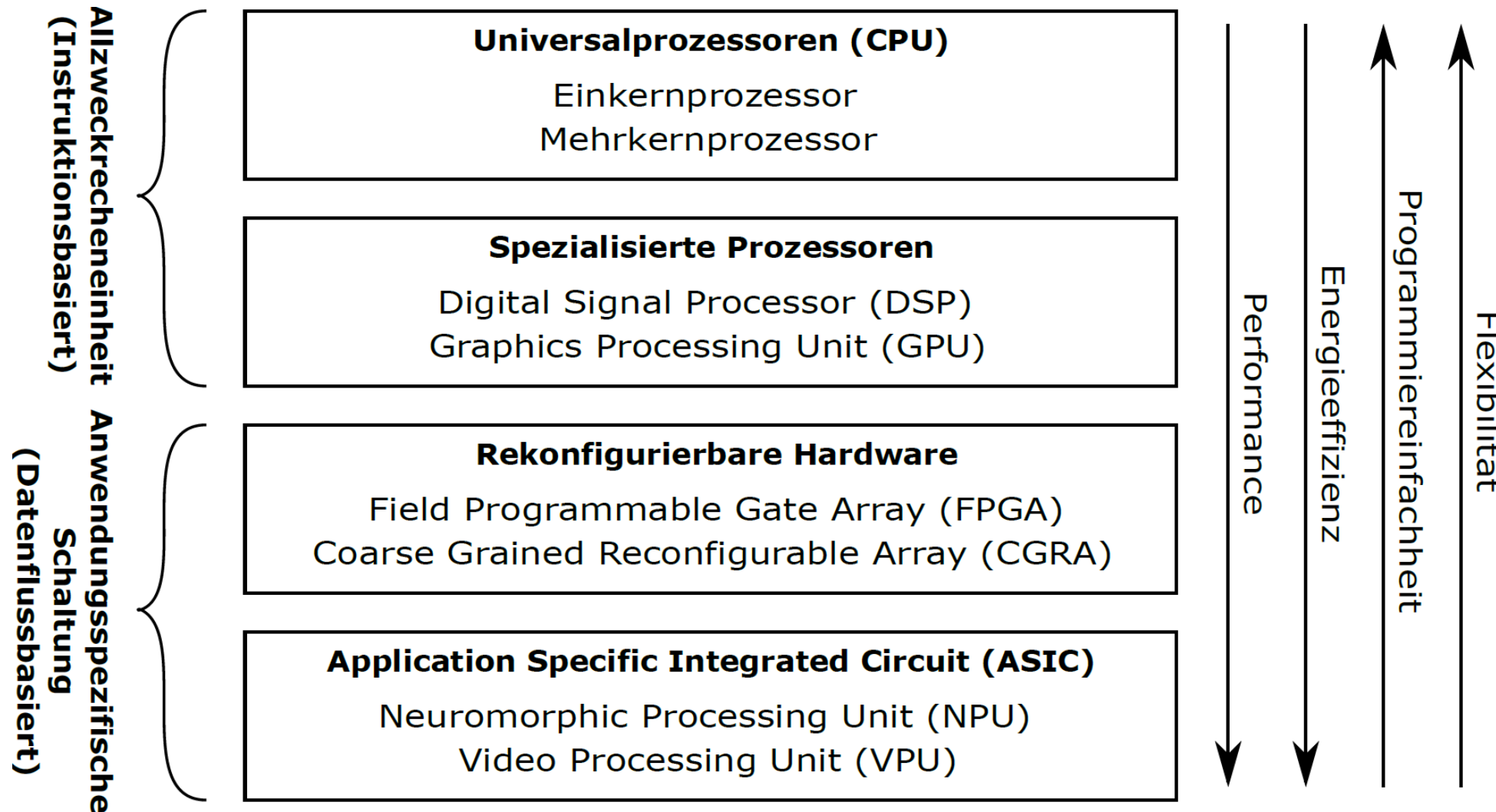
Philipp Holzinger, Dietmar Fey

---

# Motivation

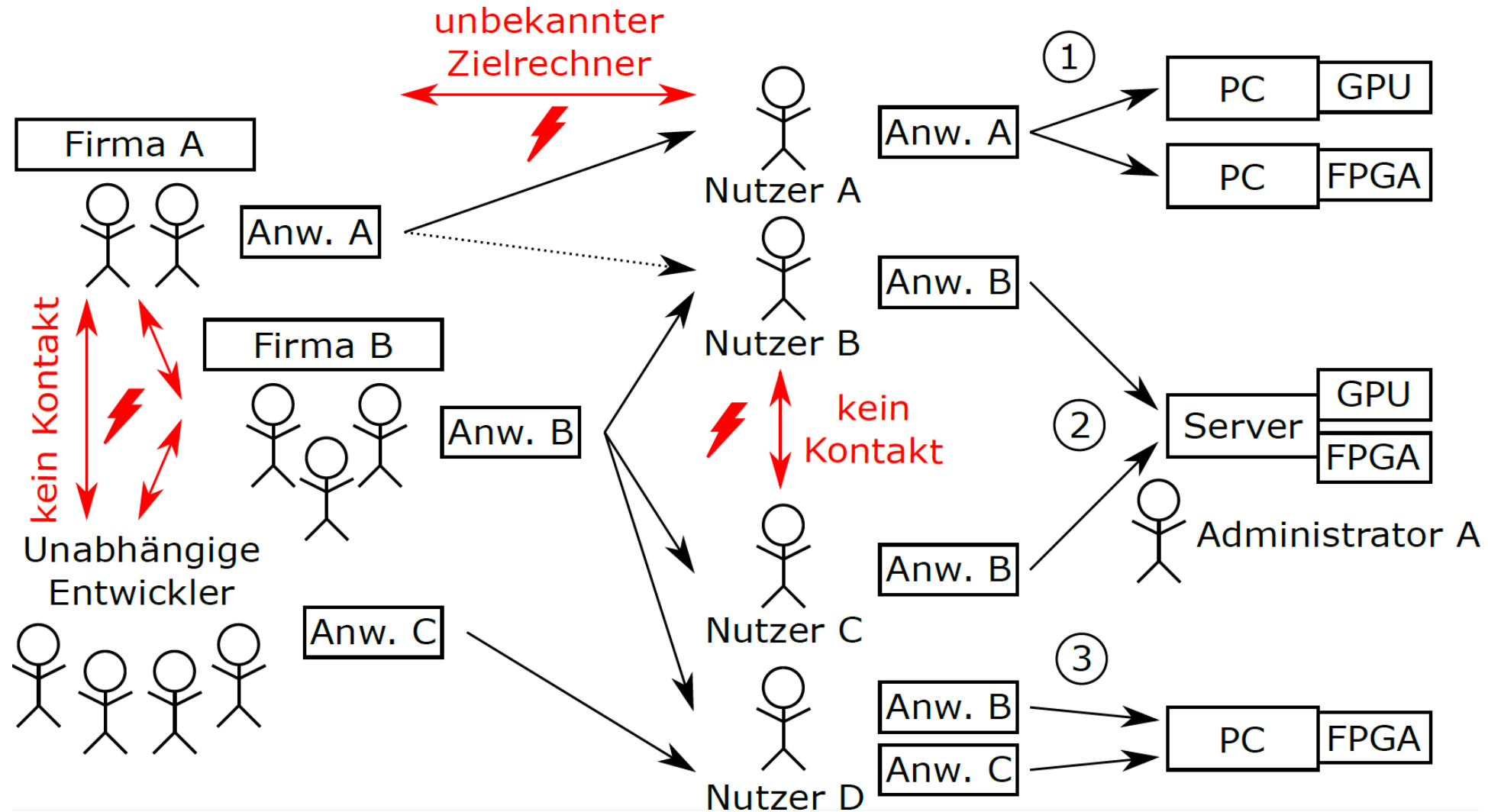
# Die HERA Methodik

Motivation - Hardwarespektrum



# Die HERA Methodik

Motivation - typisches Software-Ökosystem



- Feststellung:
  - Software-Ökosystem geprägt von **begrenztem Wissen**
  - Entwickler und Nutzer erwarten, dass Sachen „**einfach funktionieren**“ – am besten schnell / effizient
  - Funktioniert auf Universalprozessoren sehr gut
  - Zu geringerem Grad auch bei GPUs (Stichwort: CUDA auf nicht-Nvidia GPUs)
- FPGAs jedoch:
  - **Schwierig zu Programmieren** (Hardwareentwicklungssprachen / HLS mit vielen spezifischen Pragmas)
  - HLS-Kernel-Software zwischen unterschiedlichen FPGAs **schwierig zu portieren / skalieren**
  - Vergleichsweise wenige im Markt vorhanden (warum Mühe machen?)

---

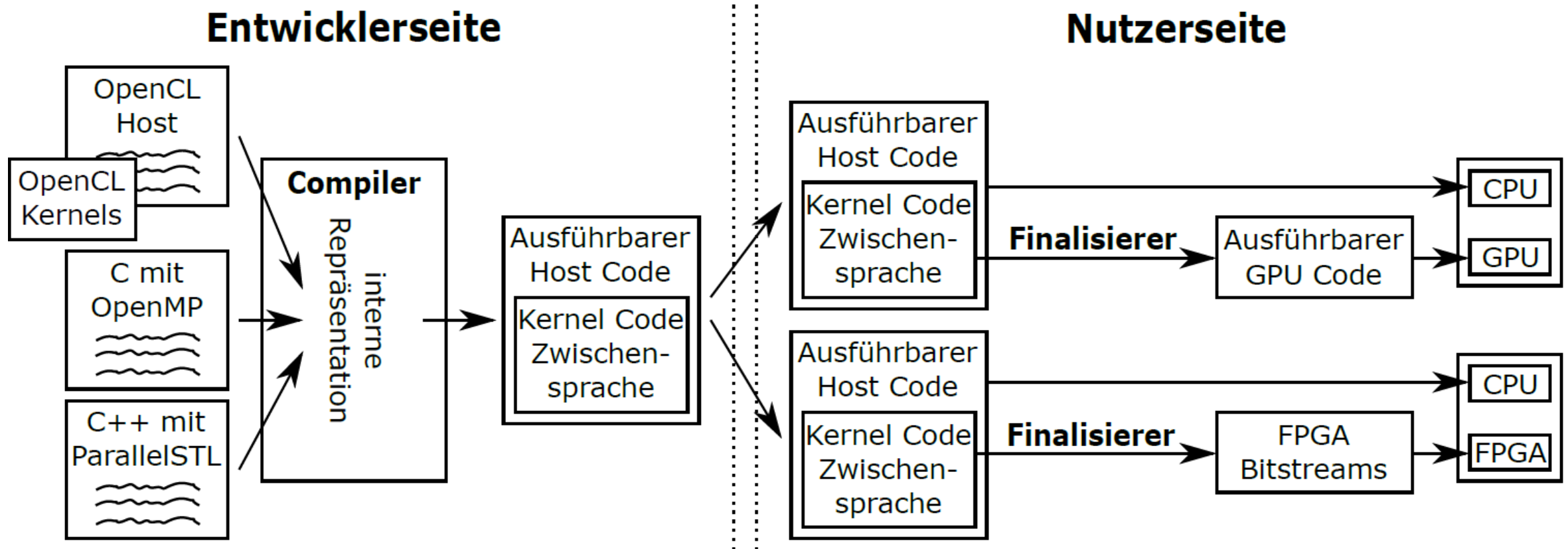
# HERA Konzept

# Die HERA Methodik

Konzept - Übersetzung der Software



Rückgriff auf bekanntes Konzept der Übersetzung in eine hardwareunabhängige Zwischensprache (wie z.B. SPIR-V)

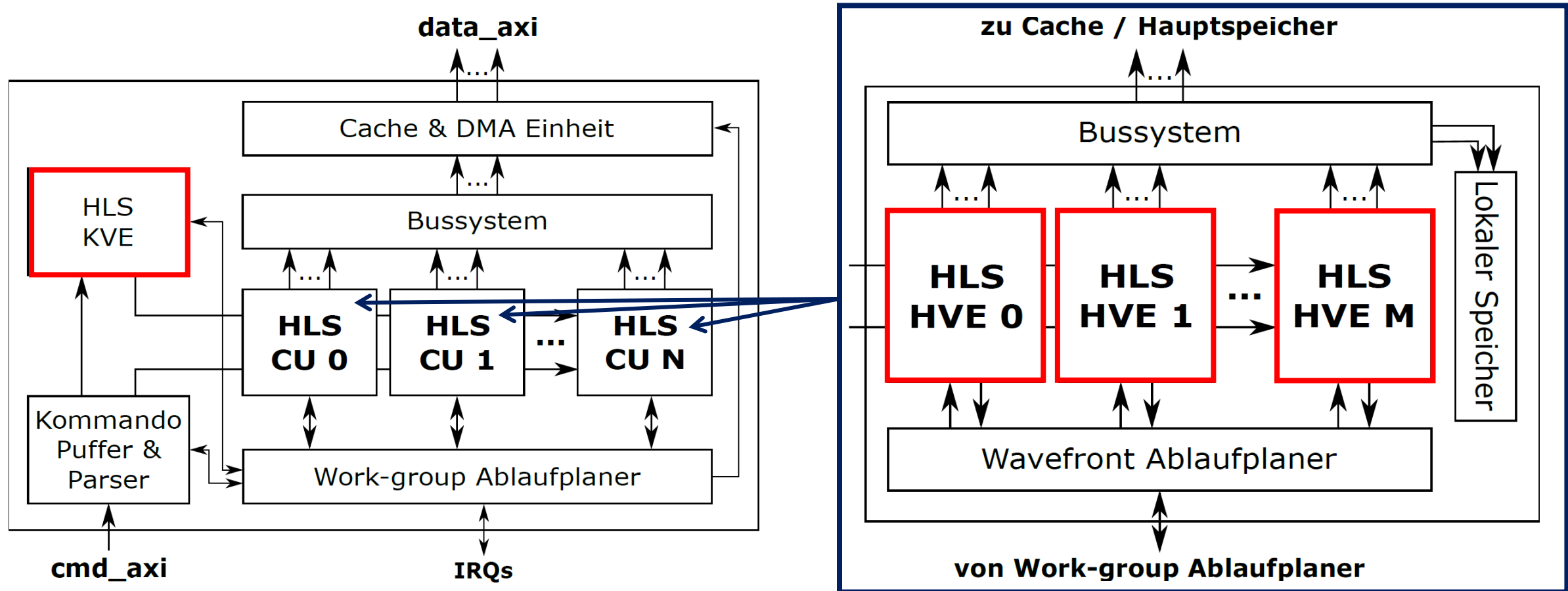


# Die HERA Methodik

Konzept – High Level Synthese



Hardwareunabhängige Zwischensprache → **Verarbeitungseinheiten (VE)** für statische Architektur (ggf. Hard-IP)



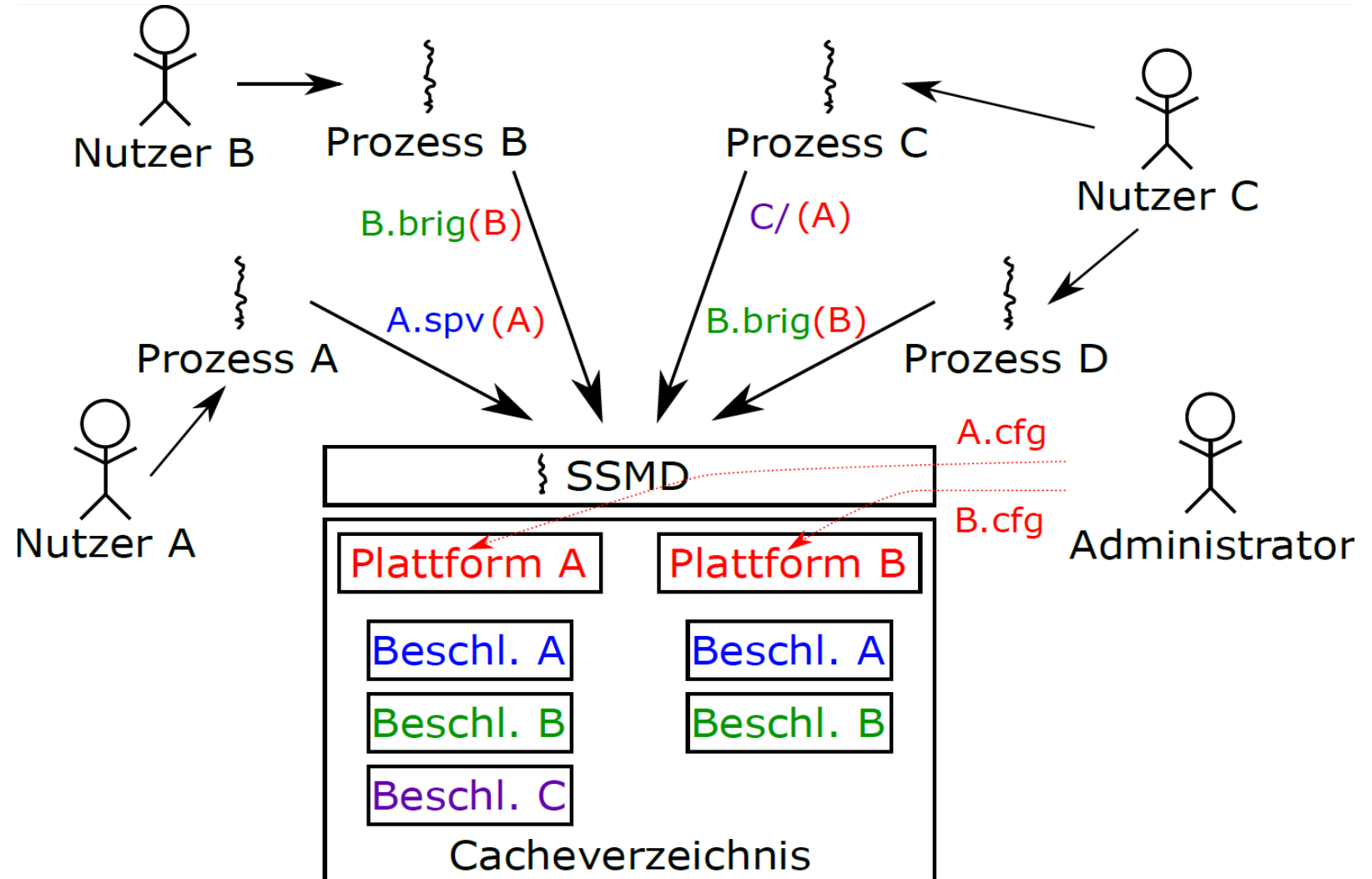
HVE = Haupt-VE / KVE = Konstanten-VE

# Die HERA Methodik

Konzept – Systemdienst zur automatischen DSE und Synthese



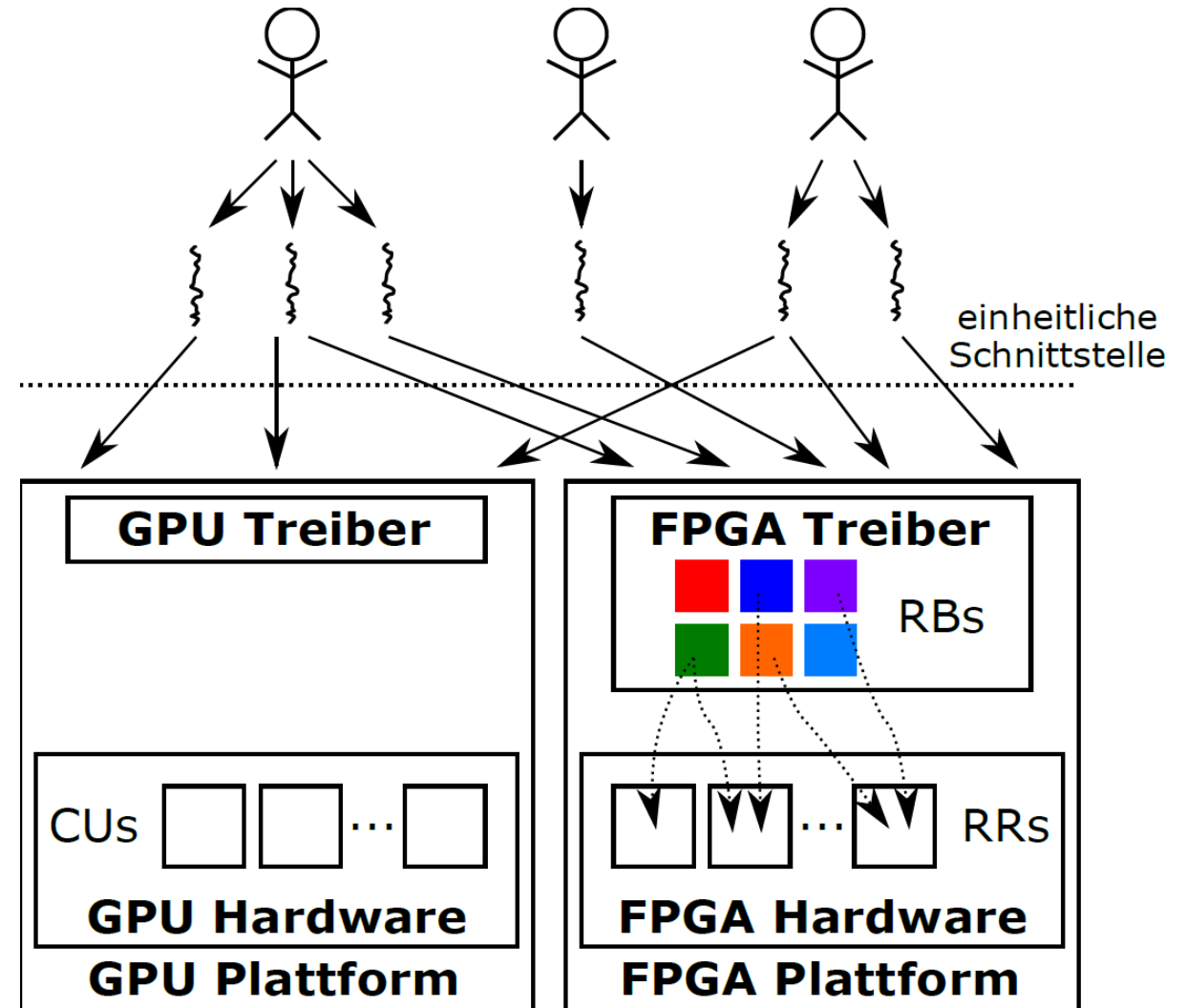
- 1) Wenn FPGA Overlay:  
Administrator konfiguriert Plattform nach Bedarf (#Beschleuniger, #Kerne, Größe der Regionen)
- 2) Installer / Anwendung fragen ausführbare Handles zu Kernels beim Systemdienst (SSMD) an
- 3) SSMD übernimmt selbstständig HLS, DSE, Synthese und Caching



# Die HERA Methodik

Konzept – nebenläufige Ausführung und automatische Rekonfiguration

- Einheitliche Schnittstelle / API für alle Arten von Beschleunigern (hier: Integration in AMD ROCm)
- FPGAs müssen nebenläufig von mehreren Prozessen nutzbar sein (Virtualisierung der Ressourcen)
- Automatische, dynamische Einplanung der Kernels und Rekonfiguration der benötigten Beschleuniger (RB)



---

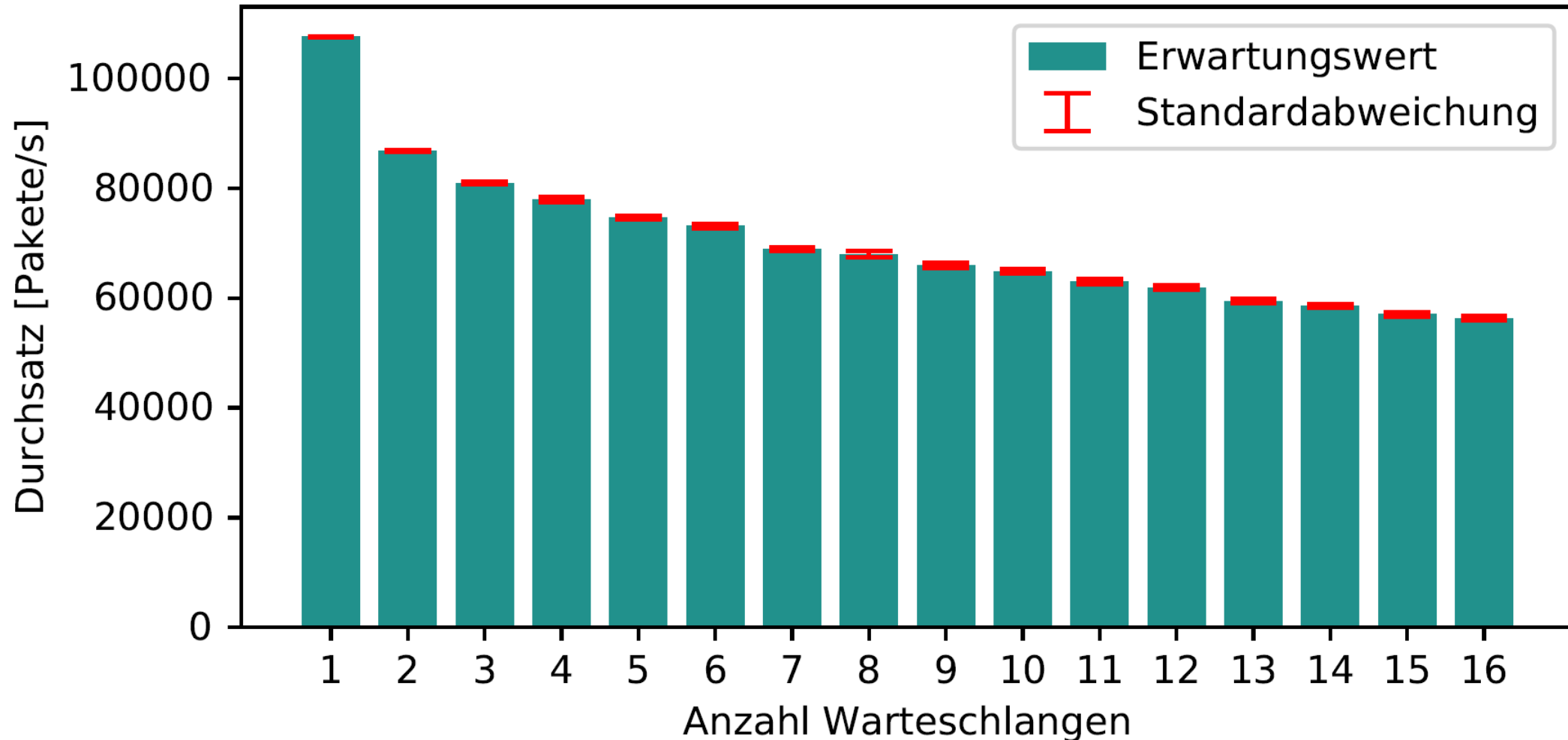
# Ergebnisse des FPGA Prototyps

# Die HERA Methodik

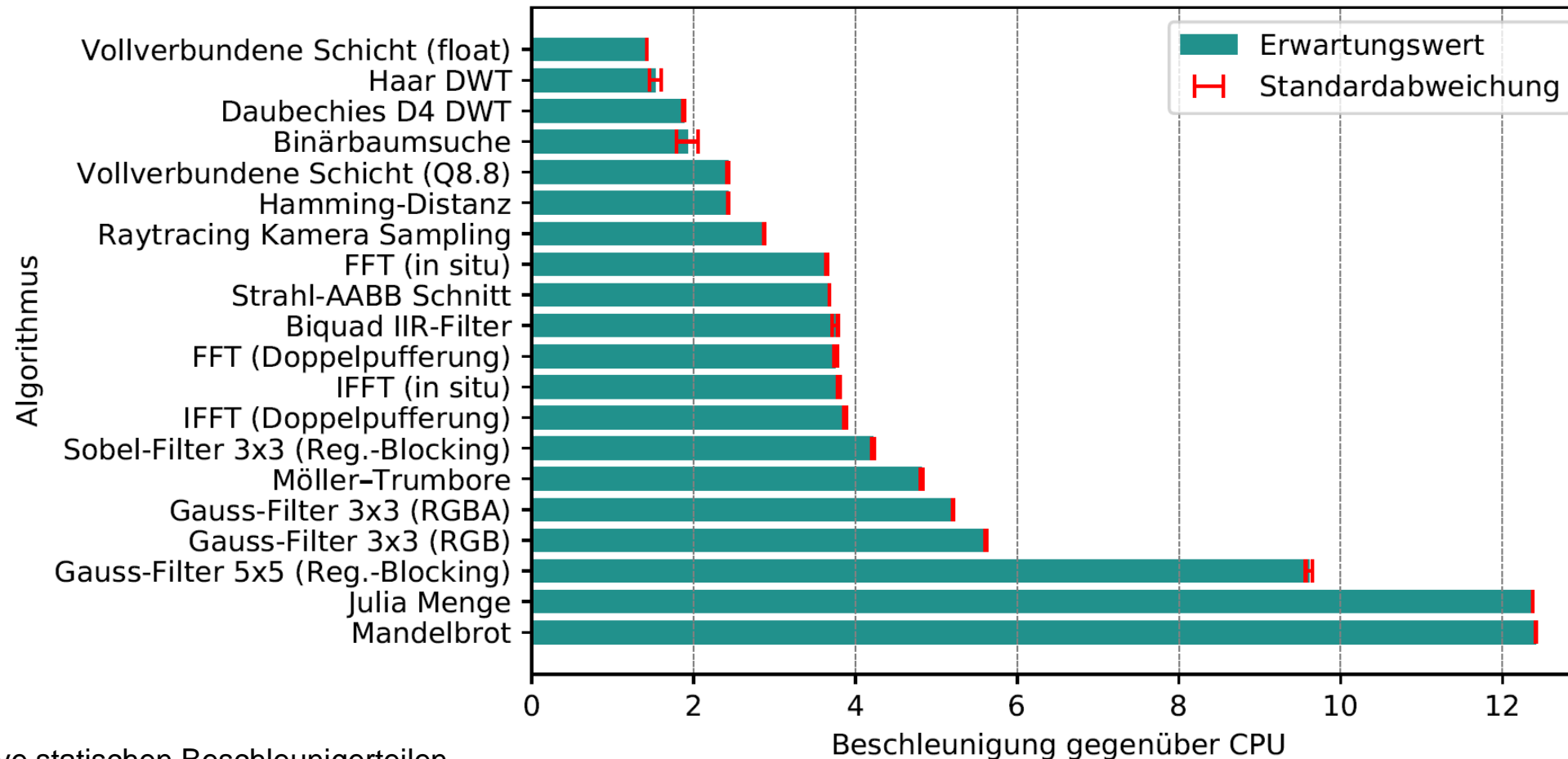
Ergebnisse des FPGA Prototyps



Overhead des HERA Frameworks (gemessen mit leeren Kernels auf Ultra96 Board)



### ARM Cortex A53 (1,2 GHz) – HERA OpenCL FPGA (214 MHz) (15k-20k LUT\*)



\*inklusive statischen Beschleunigerteilen

---

# Schlussfolgerungen

- Overlay Architekturen generell wichtig
  - jedoch **umfangreichere Hard-IP Komponenten** für bessere Performance / Energieeffizienz notwendig
- Umfassende **Virtualisierung der FPGA Ressourcen** notwendig
  - Anwendungsentwicklung ohne Kenntnisse über genaue Menge an Ressourcen
  - Anwendungen dürfen FPGA Nutzung nicht gegenseitig blockieren können
- Einsatz **hardwareunabhängiger Zwischensprachen unverzichtbar** für mehr Heterogenität
- Große Notwendigkeit **eigenständiger arbeitender / optimierender HLS-Tools**

# Vielen Dank für Ihre Aufmerksamkeit!

P. Holzinger and M. Reichenbach, "The HERA Methodology: Reconfigurable Logic in General-Purpose Computing", in IEEE Access, vol. 9, pp. 147212-147236, 2021, doi: [10.1109/ACCESS.2021.3123874](https://doi.org/10.1109/ACCESS.2021.3123874)