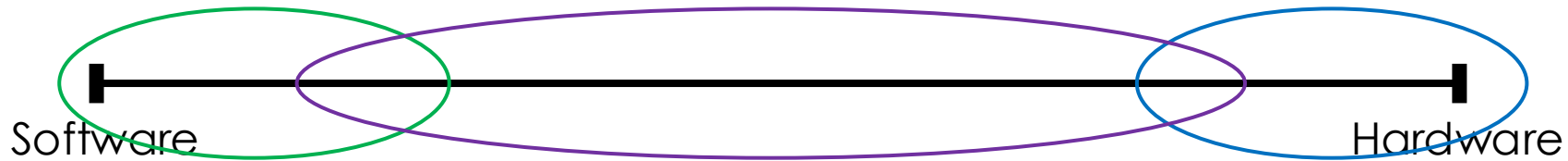


# Related work: software-hardware co-design grouped by the customisation target



## Transparent compilers, portable and vendor-specific parallel programming models, standard compilers:

- Significant promise in terms of code portability (except vendor-specific models) and performance programming facilitation
- No hardware optimisations
- Explicit accelerated code needed, potentially with a different paradigm than the rest of the code (except transparent compilers)
- Lack of the bigger picture
- Limited language support
- Limited automatic hardware selection / automatic accelerated code selection

## Software-hardware co-design frameworks and modular compilers:

- The best of both worlds: effortless programming, the potential for optimising both software and hardware, better scalability across languages
- Lack of the bigger picture
- Limited automatic hardware selection / automatic accelerated code selection

↑  
**Adaptyst will bridge the gaps here!**

## High-level synthesis:

- Takes hardware optimisations into account
- Assuming that custom hardware must be generated for an entire code
- Lack of the bigger picture
- Hardware-like thinking required
- Limited language support

**Bibliography:** See references made in section 4 of [the Adaptyst introductory paper on arXiv](#).

# Related work: design space exploration

- In terms of design space exploration, a variety of methods have been investigated before [1, 2, 3, 4, 5, 6, 7, 8, 9] with a high applicability potential for the project.
- However, the work there has limitations in terms of supported workflows, hardware, and/or optimisations, for example:
  - high-level synthesis only
  - dataflow programs only
  - approximate computing only
  - CPU-FPGA or CPU-GPU systems only
  - hardware architecture design only
  - software-side or software-to-hardware mapping optimisations only

## Bibliography:

1. Sepide Saeedi et al. "A Survey on Design Space Exploration Approaches for Approximate Computing Systems". In: *Electronics* 13.22 (2024), p. 4442.
2. Benjamin Carrion Schafer and Zi Wang. "High-level synthesis design space exploration: Past, present, and future". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.10 (2019), pp. 2628–2639.
3. Meena Belwal and TSB Sudarshan. "A survey on design space exploration for heterogeneous multi-core". In: *2014 International Conference on Embedded Systems (ICES)*. IEEE. 2014, pp. 80–85.
4. Malgorzata Michalska et al. "High-precision performance estimation for the design space exploration of dynamic dataflow programs". In: *IEEE Transactions on Multi-Scale Computing Systems* 4.2 (2017), pp. 127–140.
5. Jeronimo Castrillon, Rainer Leupers, and Gerd Ascheid. "MAPS: Mapping concurrent dataflow applications to heterogeneous MPSoCs". In: *IEEE Transactions on Industrial Informatics* 9.1 (2011), pp. 527–545.
6. Andy D Pimentel, Cagkan Erbas, and Simon Polstra. "A systematic approach to exploring embedded system architectures at multiple abstraction levels". In: *IEEE transactions on computers* 55.2 (2006), pp. 99–112.
7. Maxime Pelcat et al. "An open framework for rapid prototyping of signal processing applications". In: *EURASIP journal on embedded systems* 2009.1 (2009), p. 598529.
8. Constantino Gómez et al. "Design space exploration of next-generation HPC machines". In: *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE. 2019, pp. 54–65.
9. Lana Scravaglieri et al. "Compiler, Runtime, and Hardware Parameters Design Space Exploration". In: *IPDPS 2025-39th IEEE International Parallel and Distributed Processing Symposium*. 2025.