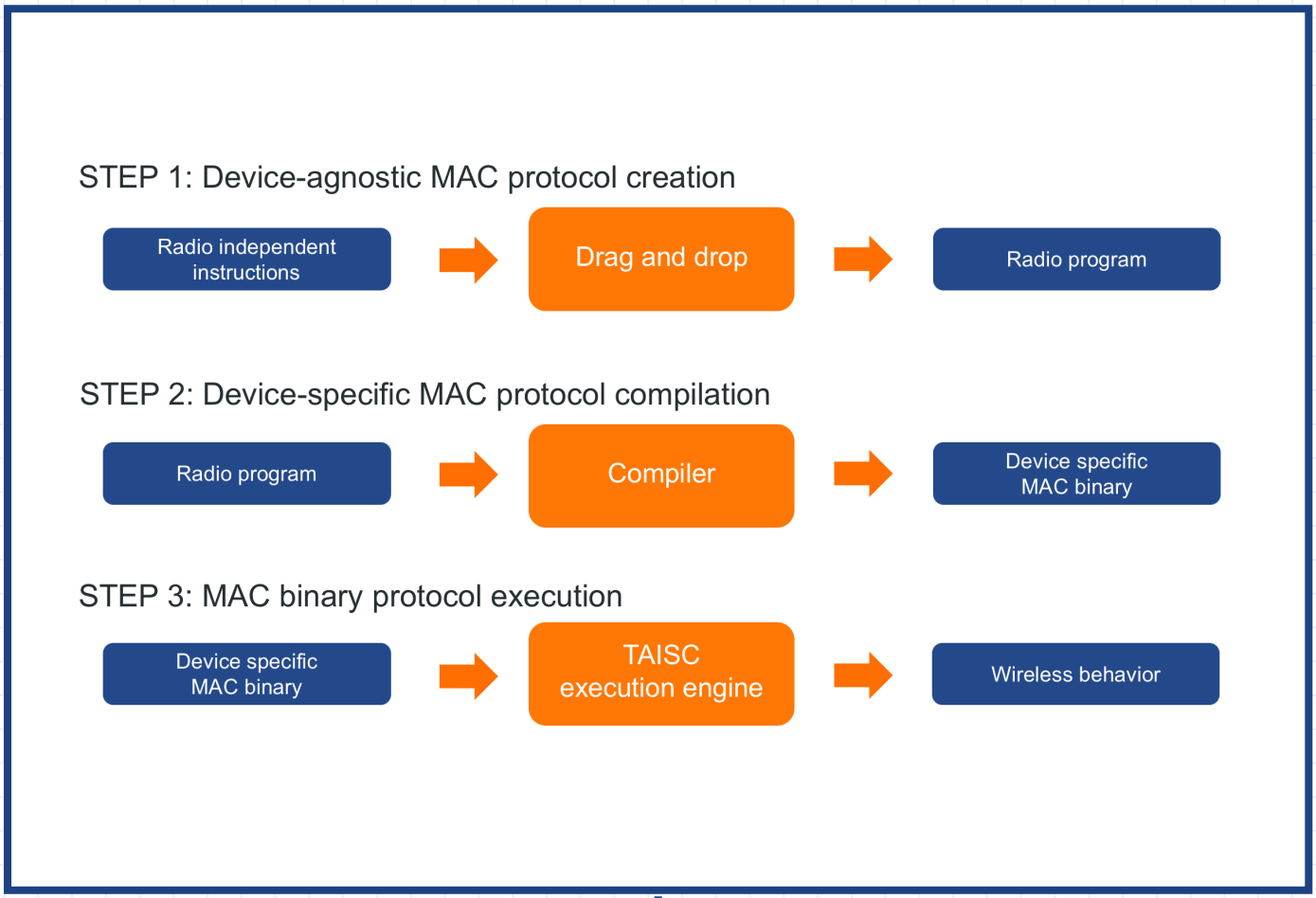
Orca

Orchestration and Reconfiguration Control Architecture

SDR DATA PLANE FUNCTIONALITY

Design-time composition: Design-time composition of MAC protocol



- MAC protocols are written as chains of device independent instructions. By consequence protocol developers don't need deep knowledge about physical layer implementation.
- The instructions are translated into device specific instructions. The resulting binary can be deployed on the desired platform.

SDR DATA PLANE FUNCTIONALITY

Design-time composition: Design-time composition of MAC protocol

CONTEXT

Current MAC protocol design requires deep knowledge about physical layer implementation. Hence, for most people, the creation of a custom MAC protocol is a tedious and long process. As current protocol implementations are hardware specific, they cannot be ported easily to new hardware platforms so they need to be rewritten numerous times. There is a need for a platform that allows for easy, platform independent MAC protocol design and implementation in a limited amount of time. As a result users will be able to prototype new protocol implementations, which can directly be used on all desired platforms. In short, we aim to offer in ORCA a MAC framework for fast protocol design using platform-agnostic abstractions, time-accurate execution and runtime control & management (for adapting and/or switching MAC protocols).

UNIQUE SELLING POINT

- MAC protocols are written using a hardware independent set of instructions. These instructions will be translated to hardware specific code by the proposed MAC platform, thus the user does not need any direct knowledge about the targeted hardware platforms and/or physical layer.
- The instructions are time annotated, meaning that for every instruction it is known in advance how long it takes to execute. This knowledge is vital in time-critical protocols where instructions can be scheduled at specific moments, to optimally use the spectrum and minimize radio energy consumption. In that way MAC engine can timely execute the user specified instructions minimizing delay and pipelining where possible operations.
- The instructions are translated (compiled) into a minimal bytecode representation. By consequence, the memory footprint per MAC protocol is smaller compared to previously available implementations.
- TAISC compiler is platform independent and hence reusable.

OPPORTUNITIES

- This offer provides a framework that can be used to rapidly prototype and test novel networking ideas on the MAC layer.
- This offer allows optimising MAC protocol behaviour by time annotation of the instructions, and translation into a minimal bytecode representation.
- This offer allows for directly porting MAC protocols to new hardware platforms with minimal effort.
- Smaller MAC protocol footprint will minimize the overhead for runtime MAC reprogramming and switching.

REFERENCES

Following git repositories consist of the Contiki tree, and the existing MAC creation framework (TAISC):

- TAISC is free of charge for academic usage, for more information please refer to [1] and http://www.wishful-project.eu/taisc

1 Jooris, B., Bauwens, J., Ruckebusch, P., De Valck, P., Van Praet, C., Moerman, I., & De Poorter, E. (2016). TAISC: A cross-platform MAC protocol compiler and execution engine. Computer Networks, 107, 315-326.