Moving Beyond Competitions:

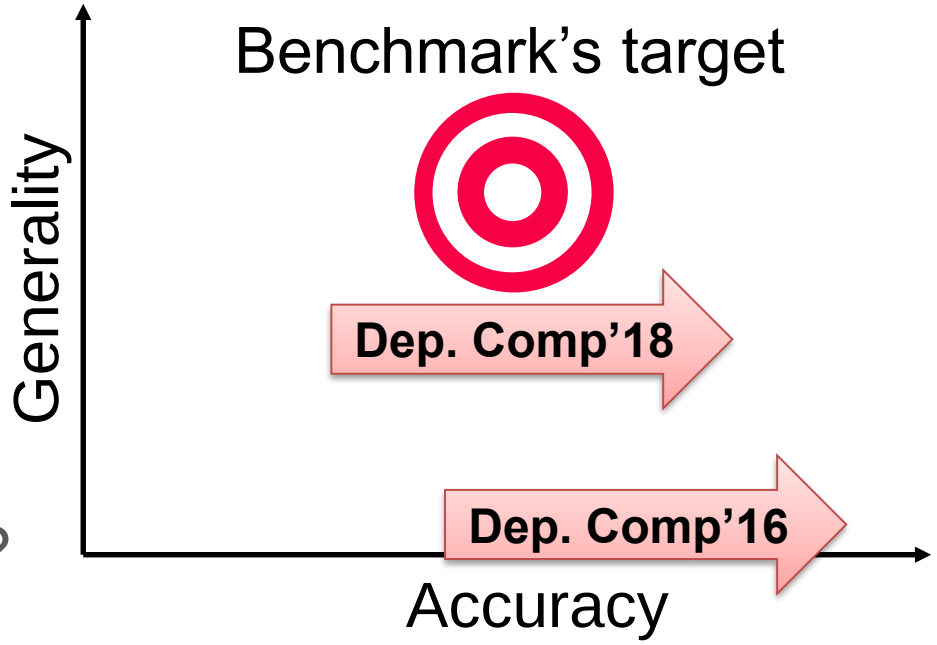
Extending D-Cube to Seamlessly Benchmark Low-Power Wireless Systems

Markus Schuß, Carlo Alberto Boano, and Kay Römer
Institute of Technical Informatics
Graz University of Technology, Austria

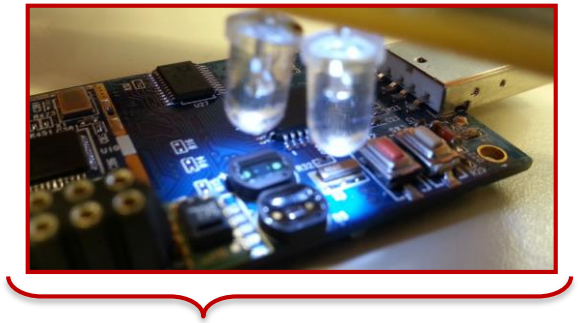
04/17/2018

Problem Statement

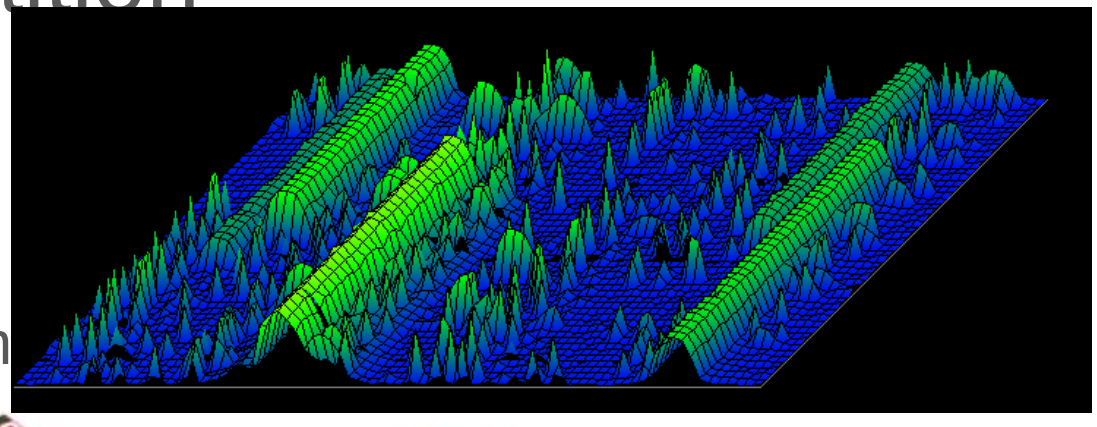
- No standardized methodology to compare the performance of low-power wireless systems
- EWSN Dependability Competition Series
  - Goal: find which low-power wireless protocol perform(s) best in harsh RF environments
  - Created a dedicated competition infrastructure (D-Cube)
- 2016: 1st edition with dense mesh network and point-to-point communication
  - Good accuracy, but no generality of results
  - Later editions became more general by focusing on multiple communication patterns
- From competition infrastructure to benchmarking infrastructure: what's missing?

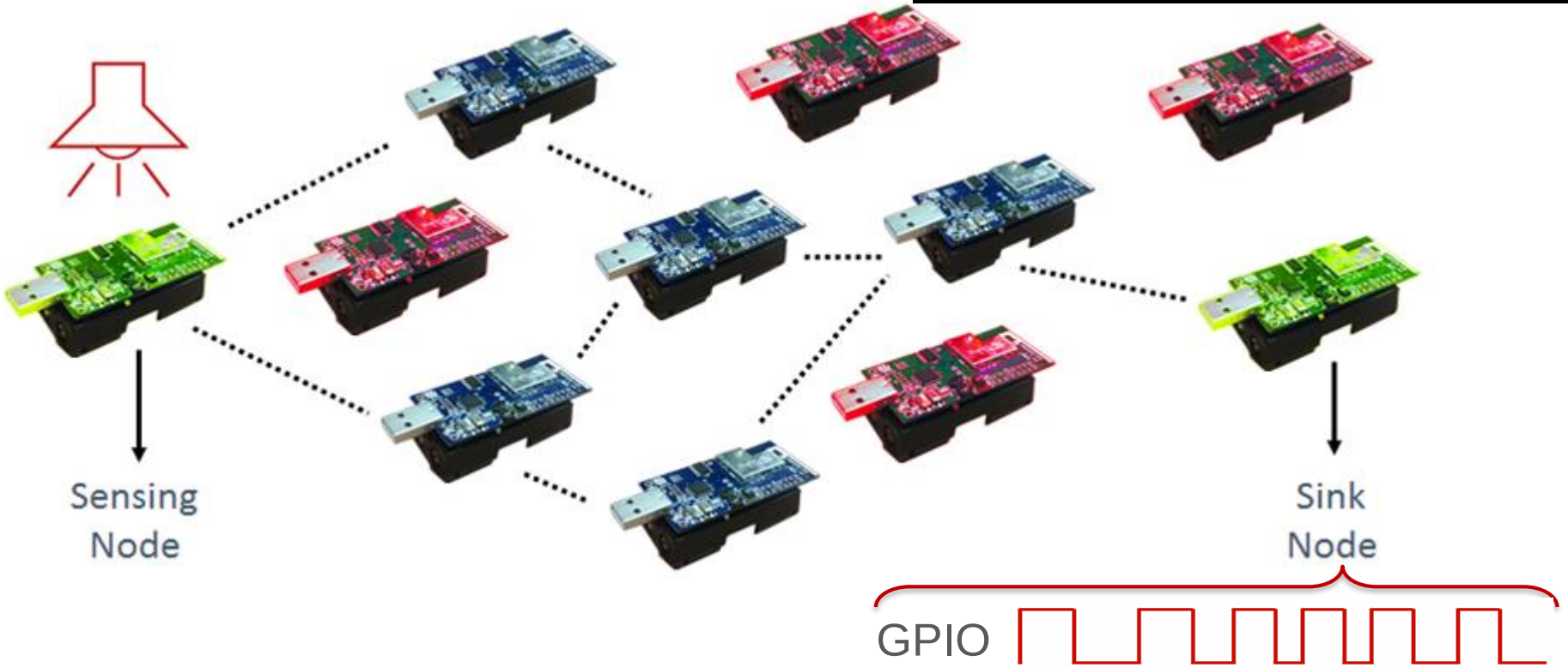


EWSN Dependability Competition



- Sensing node in (blinking LED)
- Detecting status
- Forwarding them

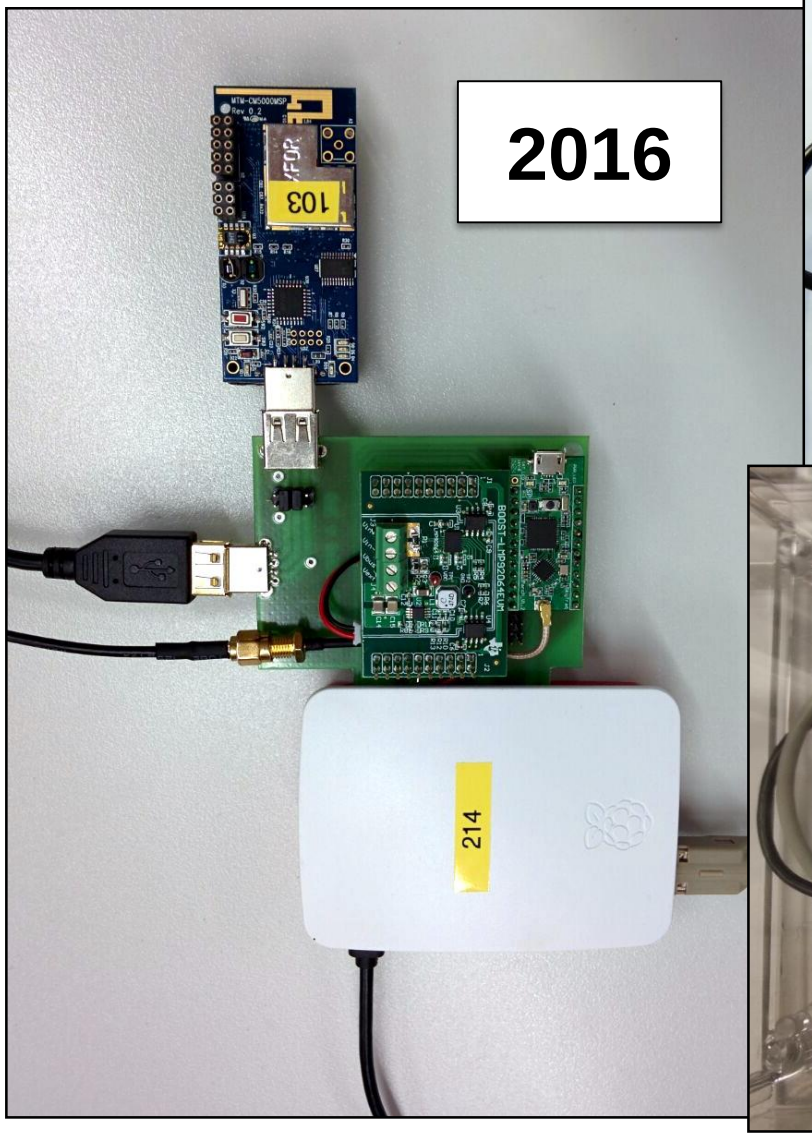


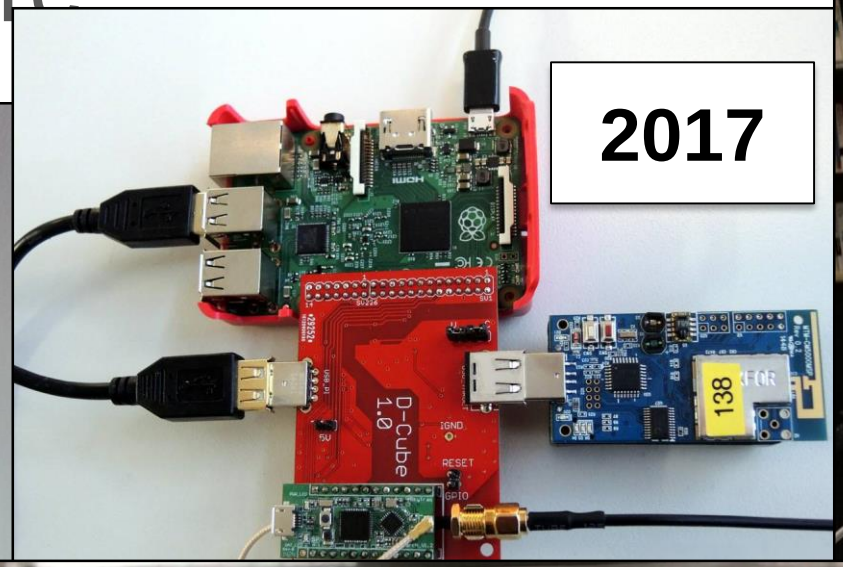


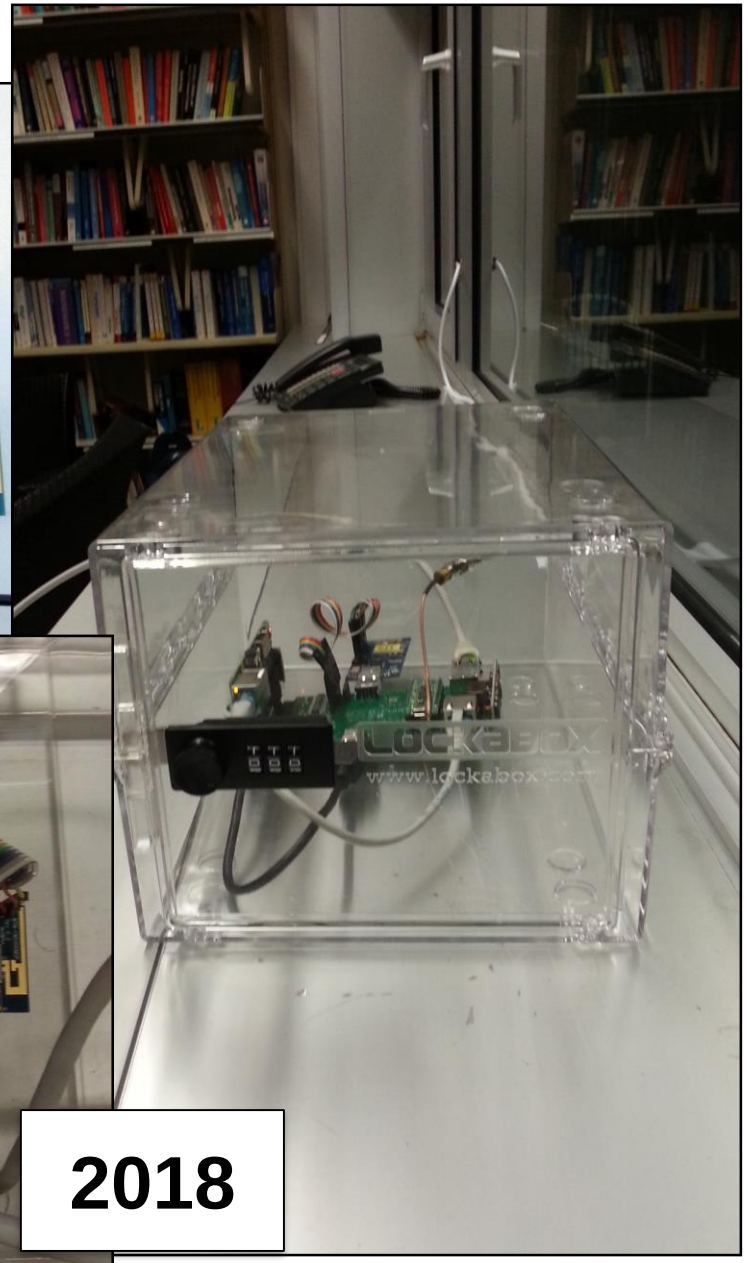
EWSN Dependability Competition

- Started as a “hackathon” (2 days prep, 1 day evaluation)
- Last iteration was ran remotely over two months (2 weeks evaluation)
- We build D-Cube, our own open, low-cost testbed infrastructure
  - Currently two instances exist, Uppsala (Sweden) and Graz (Austria)

D-Cube's hardware

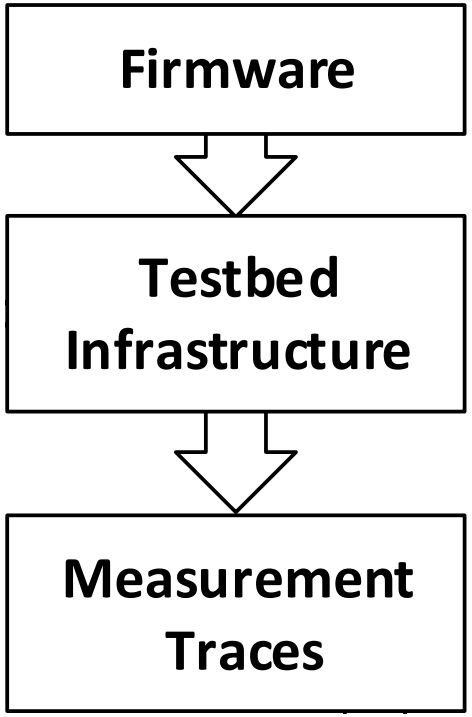




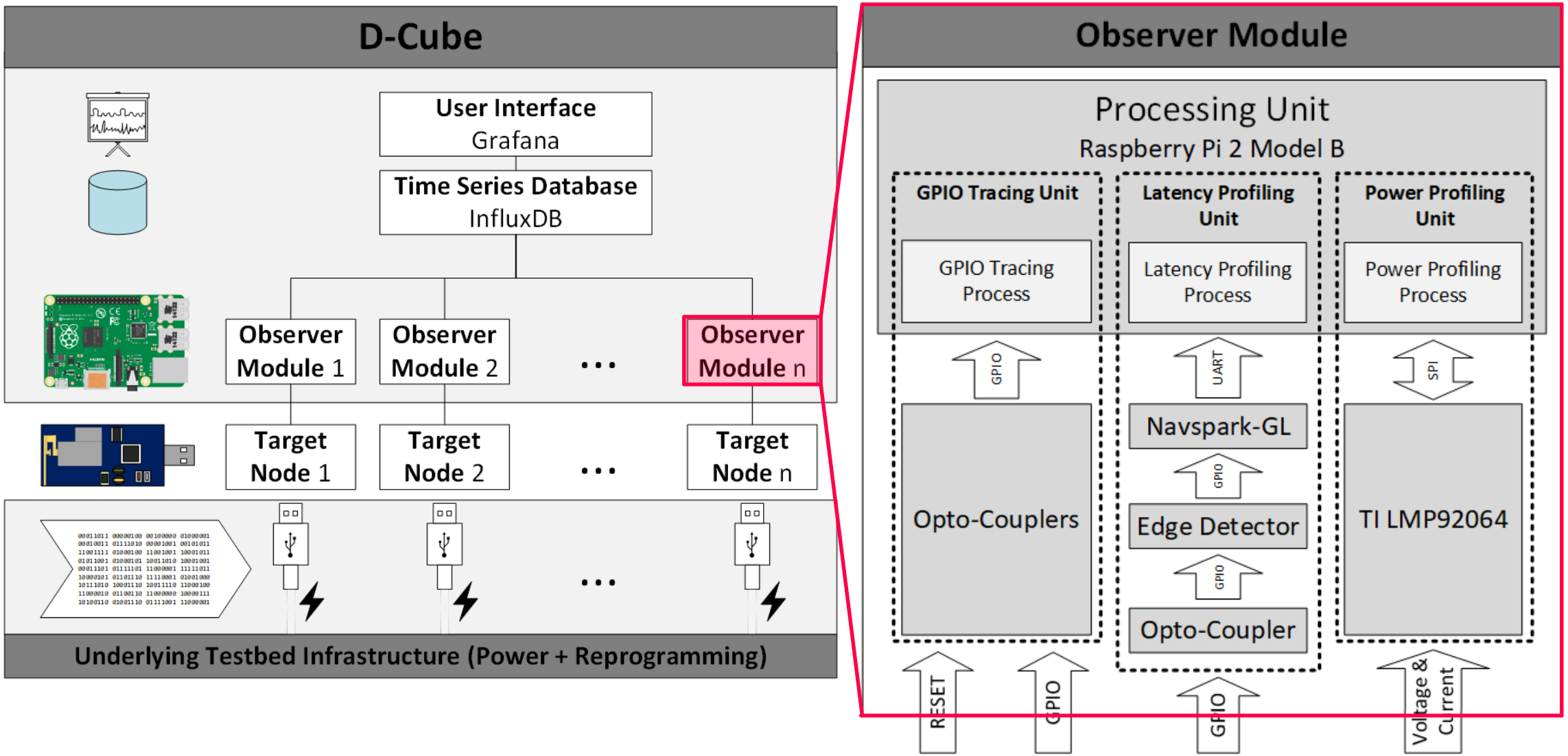


Benchmarking Requirements

- Automated, seamless and repeatable execution
  - Without modifications to the firmware
- Configurable parameters
  - Traffic pattern: point-to-point, point-to-multipoint,…
  - Traffic load: periodic, irregular, number of msg/s,…
  - System parameters: network density,…
  - Experiment parameters: number of runs, duration,…
  - Environmental parameters: RF interference, temperature,…
- Configurable metrics
  - Latency, jitter, total/peek energy consumption, reliability,…

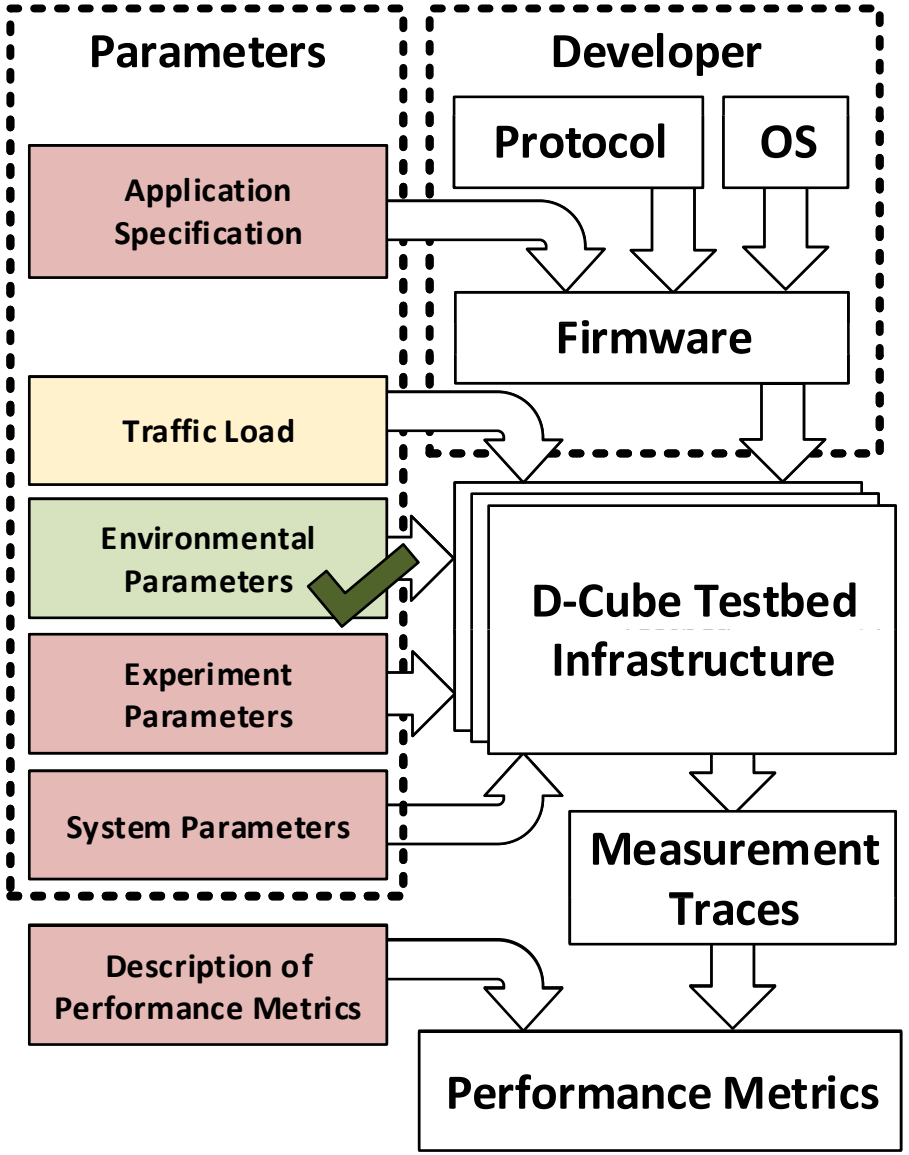


D-Cube: Architecture

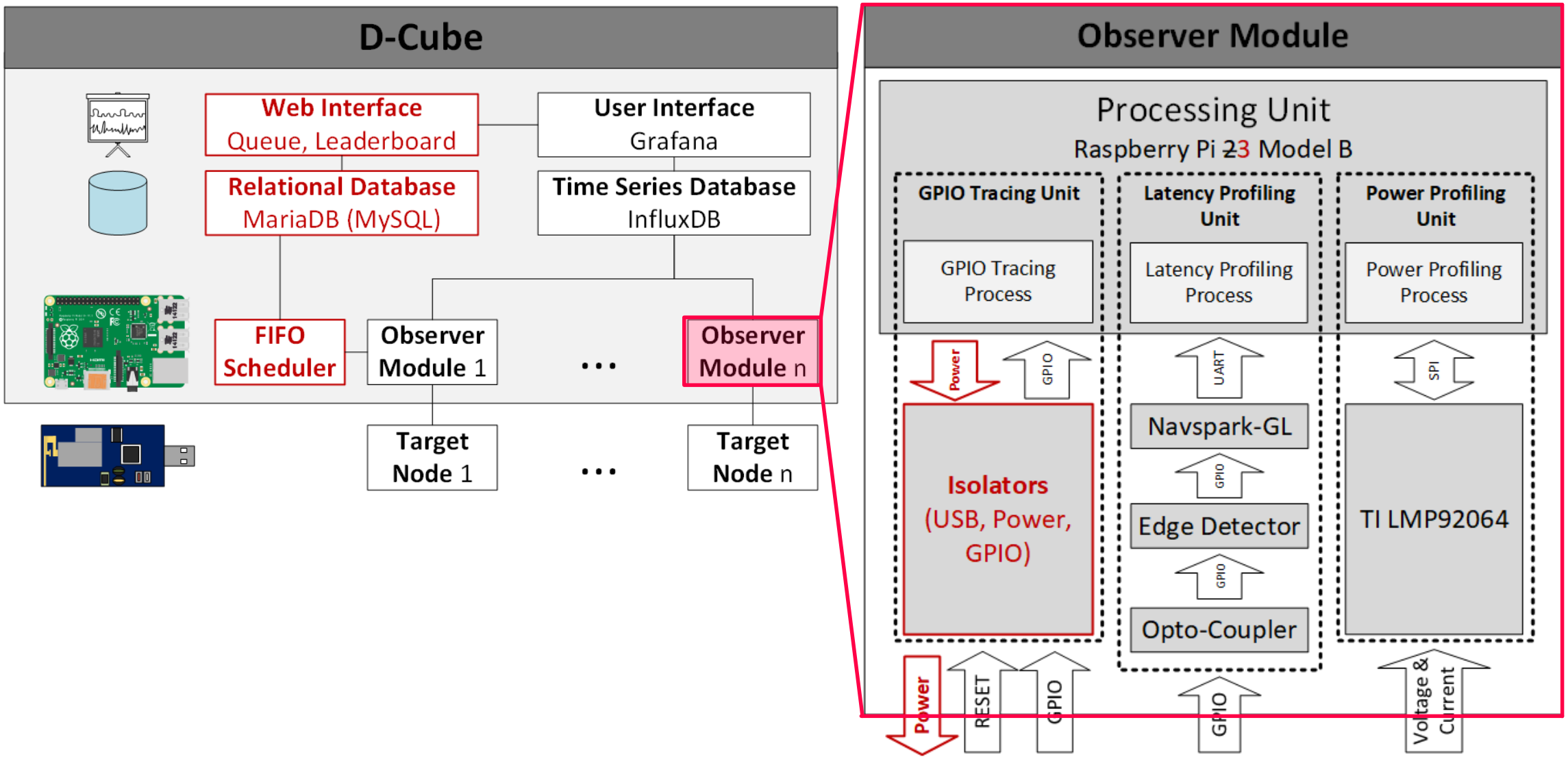


Still far-away from a benchmarking infrastructure

- Application Specification locks traffic pattern to point-to-point
- Traffic load created by a second TelosB
- JamLab integration for interference
- Manual Execution via SSH
- Static Topology

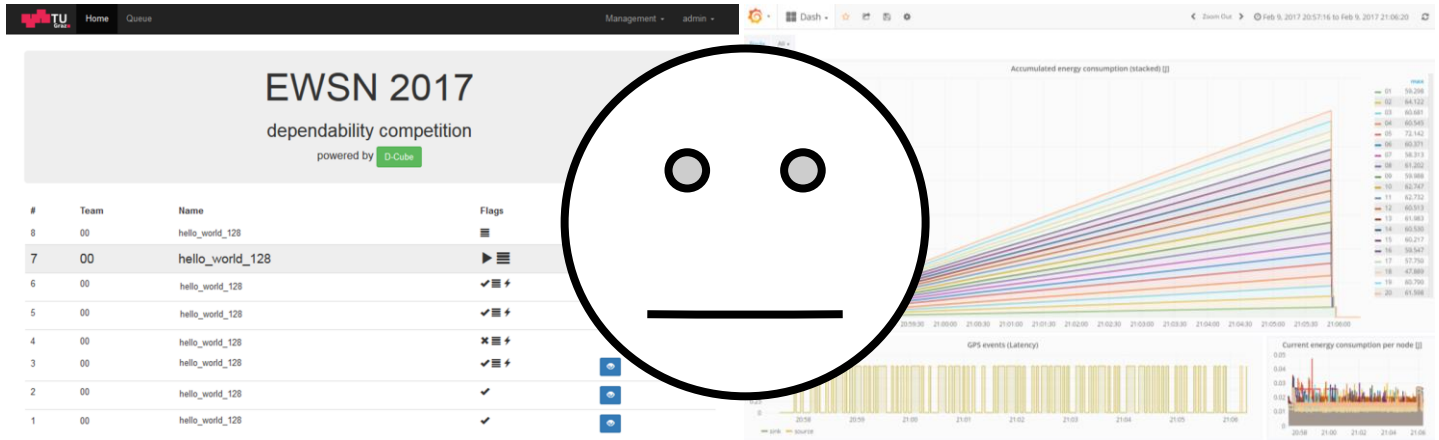


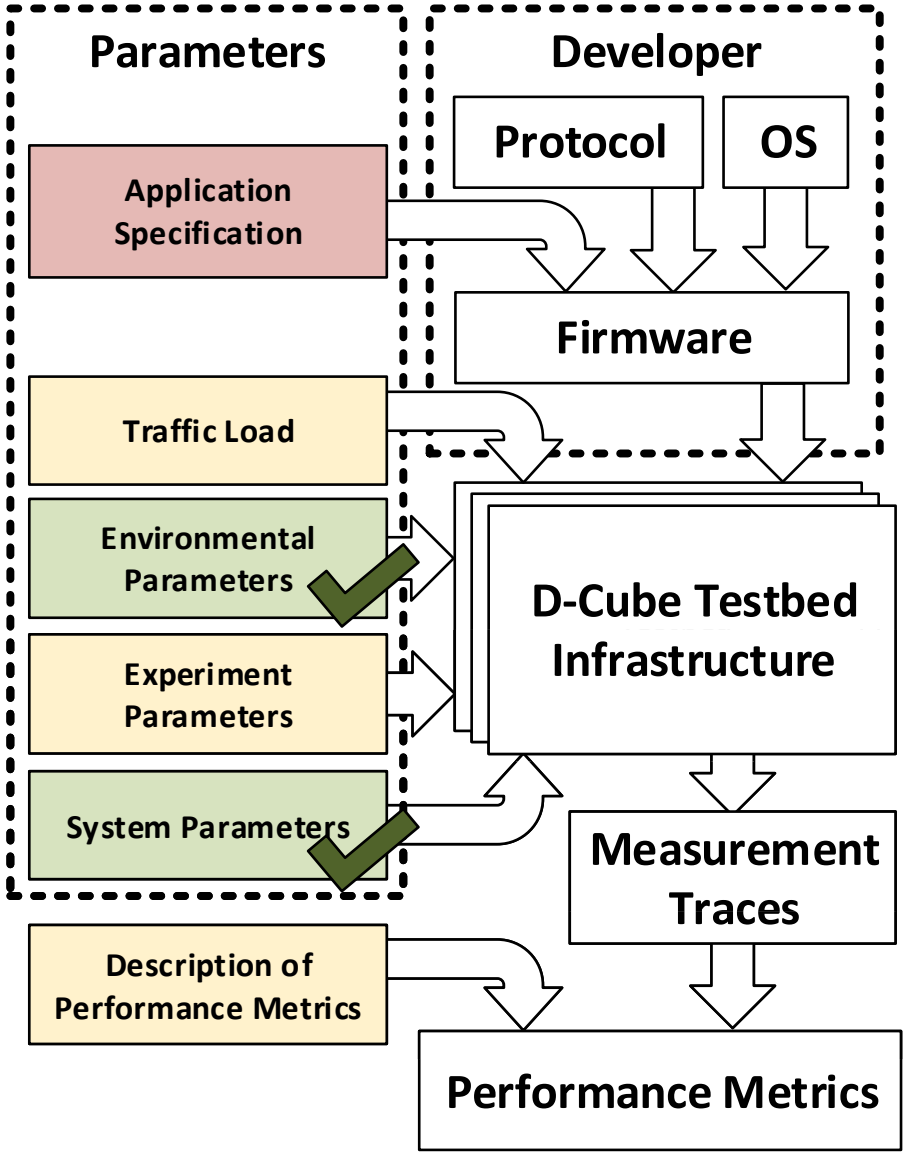
D-Cube: Architecture (first Update)



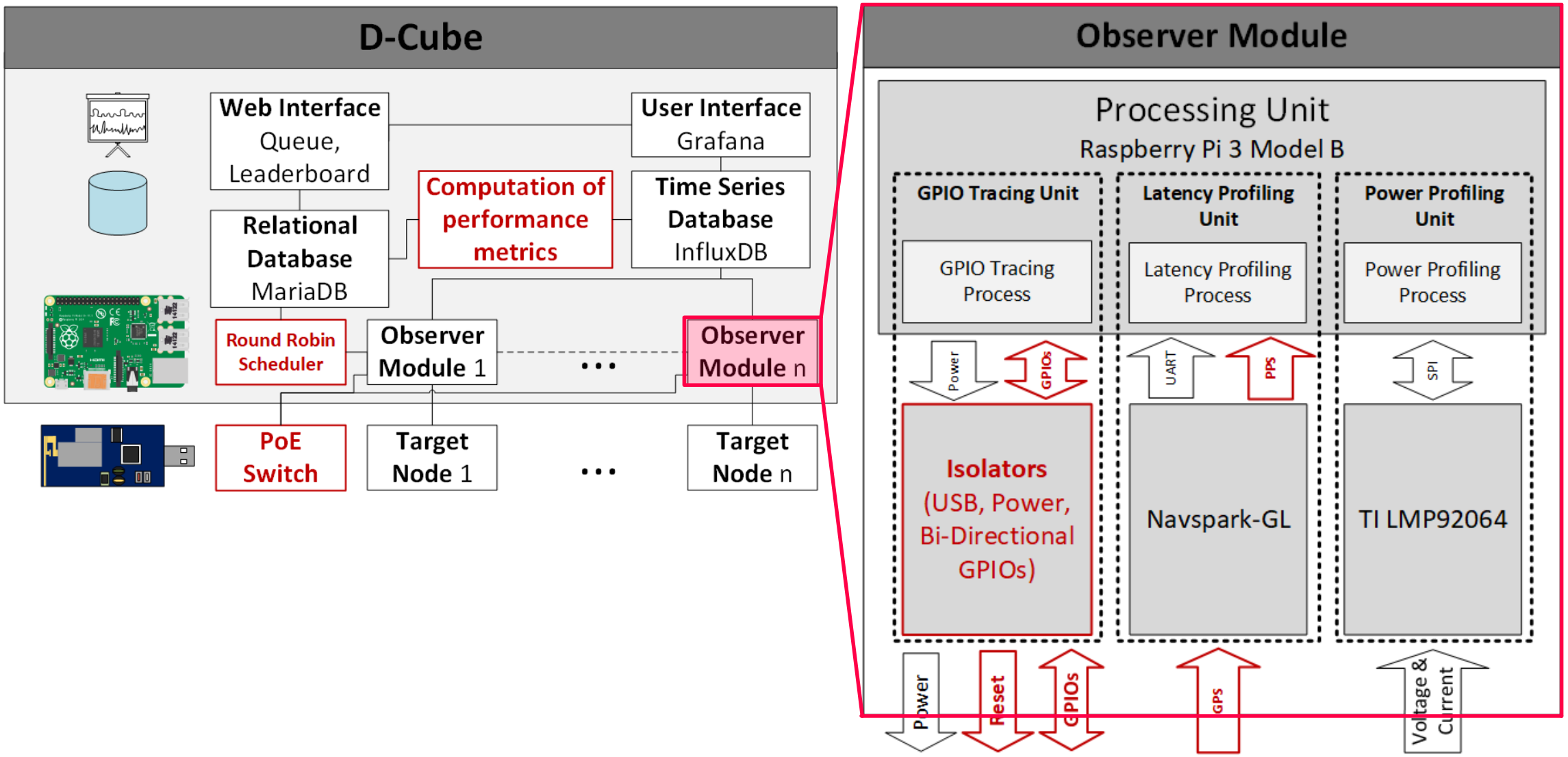
Does this work for benchmarking now?

- Management Web interface
  - Enables remote execution
  - Enables queuing of experiments
- Nodes can be turned off to simulate sparse network



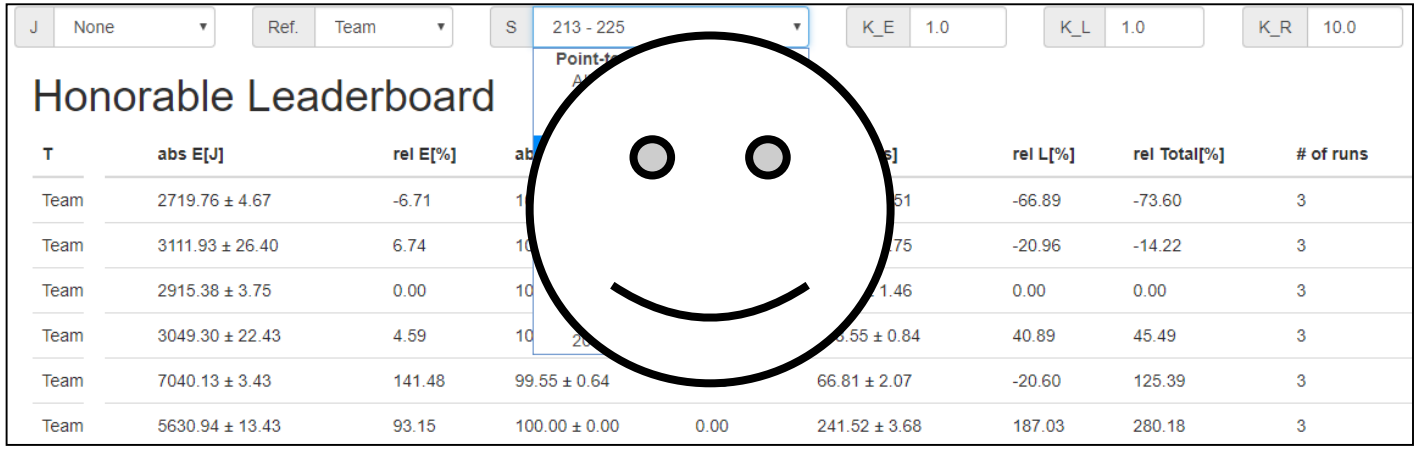


D-Cube: Architecture (second Update)



Benchmarking Infrastructure

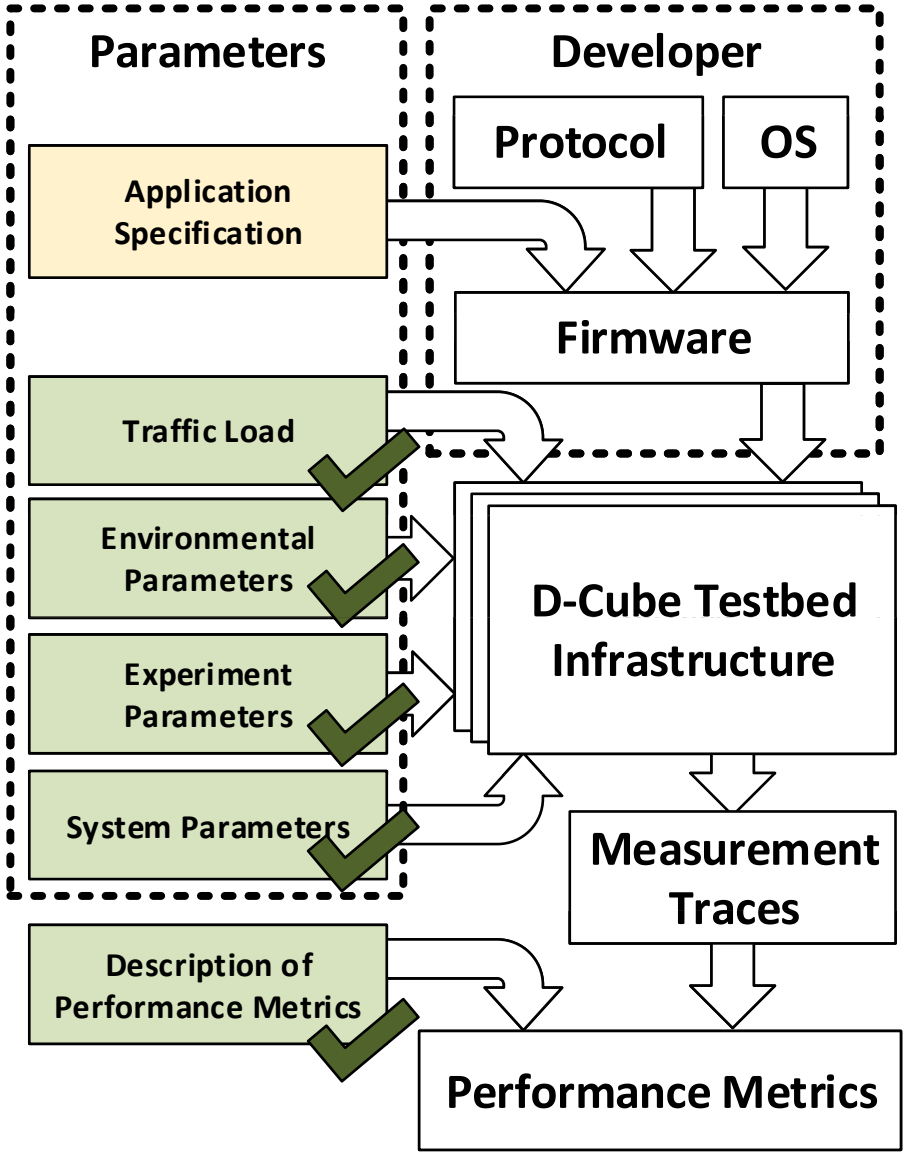
- Multiple traffic pattern in parallel
  - Available on every target node
- Traffic load fully in software now
- ReST API for automation
- Grouping and statistics available for performance metrics



J None | Ref. Team | S 213 - 225 | K_E 1.0 | K_L 1.0 | K_R 10.0

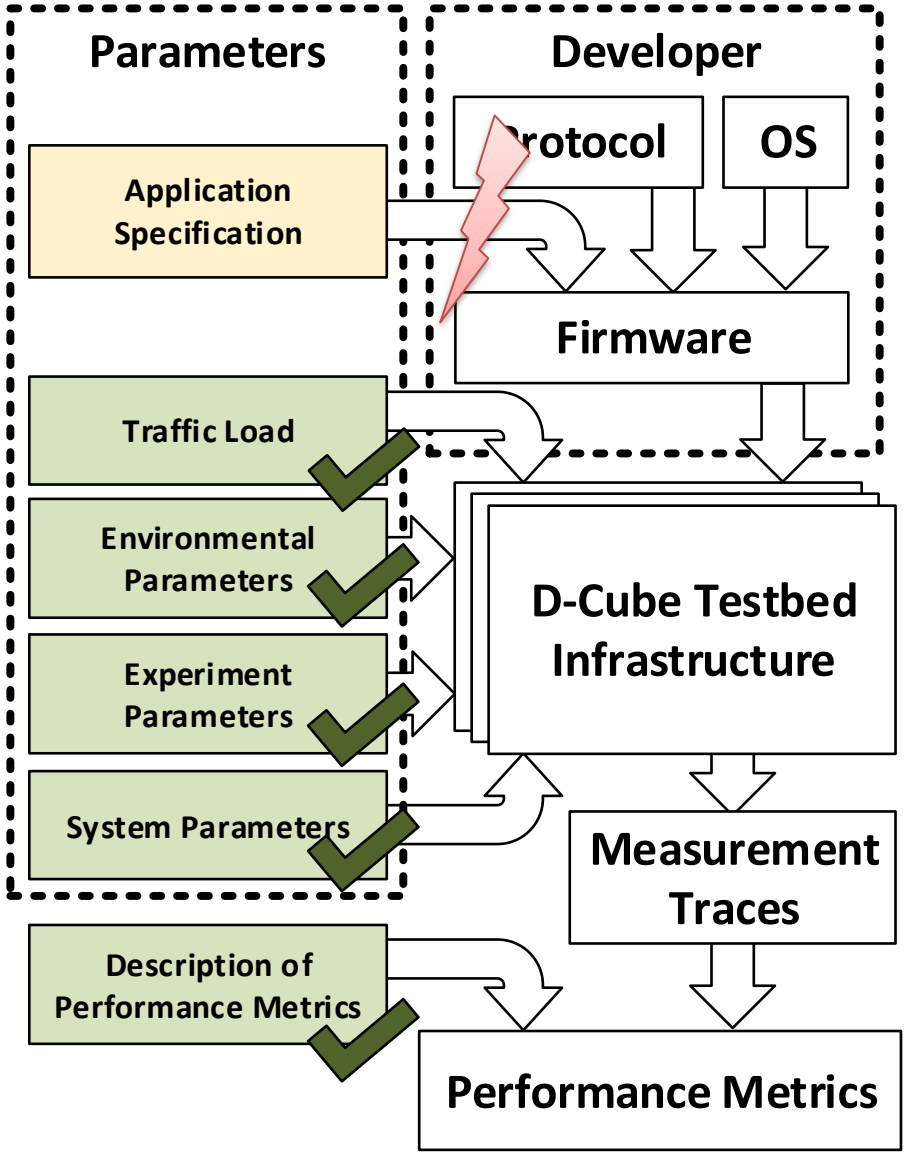
Honorable Leaderboard

T	abs E[J]	rel E[%]	ab			rel L[%]	rel Total[%]	# of runs
Team	2719.76 ± 4.67	-6.71	1		51	-66.89	-73.60	3
Team	3111.93 ± 26.40	6.74	10		75	-20.96	-14.22	3
Team	2915.38 ± 3.75	0.00	10		1.46	0.00	0.00	3
Team	3049.30 ± 22.43	4.59	10		3.55 ± 0.84	40.89	45.49	3
Team	7040.13 ± 3.43	141.48	99.55 ± 0.64		66.81 ± 2.07	-20.60	125.39	3
Team	5630.94 ± 13.43	93.15	100.00 ± 0.00	0.00	241.52 ± 3.68	187.03	280.18	3



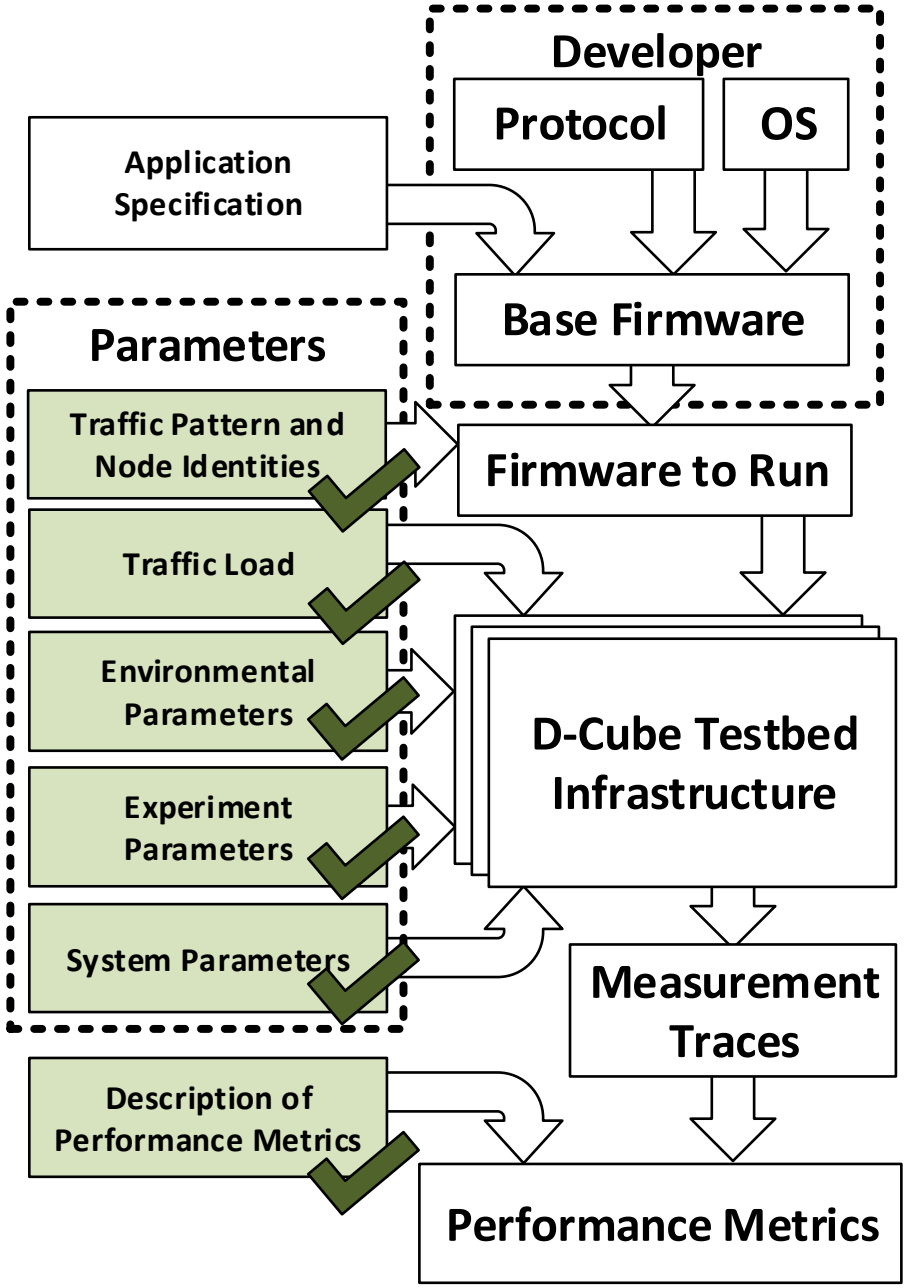
The Problem

- Traffic pattern and node identities intertwined with the specification
  - Can not be changed without access to the source or the developer
  - Node identities are fixed, which may include optimisations based on topology
  - Prevents execution on other testbeds



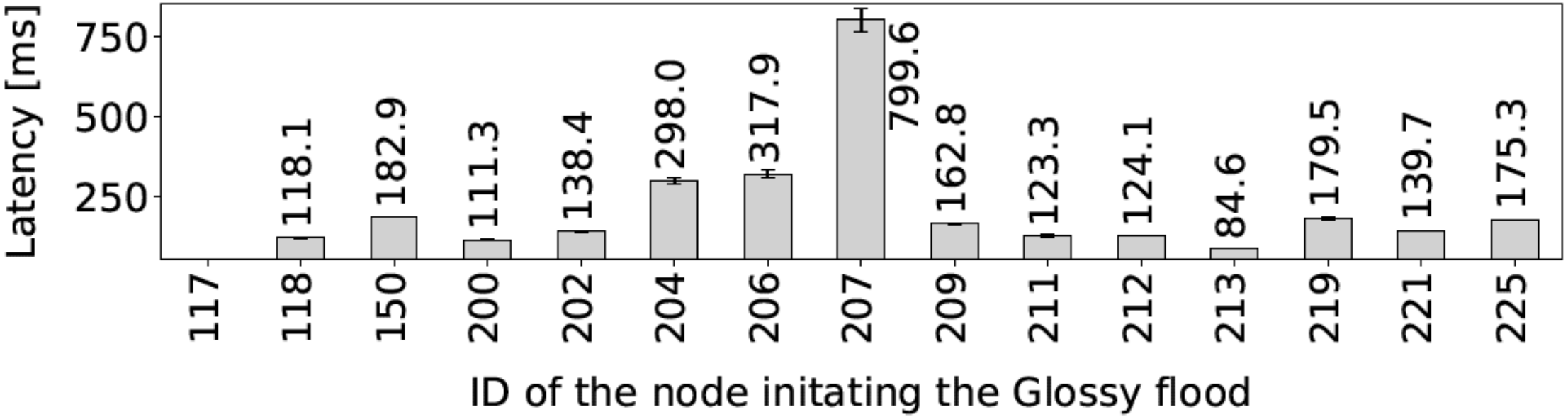
Our proposed Solution

- Split the traffic pattern and node identities from the specification
  - Define a well-known data structure containing these information
  - Developer provides their memory address
  - Values are replaced according to the parameters
  - Includes the option for user-defined variables



User-defined Variables

- Build a custom firmware from the public source
- Replaced only the user-defined variable `crystal_sink_id`
- Automatically evaluated the resulting latency



EWSN Dependability Competition 2019?

- In 2018 we compared synchronous transmissions against synchronous transmissions (mostly glossy)
  - While fun, it looks like routing (RPL) based solutions do not care
- Binary events only
  - Would an (emulated) sensor or more complex scenario (e.g. payload and destination via UART) make a difference?
- Aging target node limits solutions
  - Newer hardware lacks/prohibits implementations?

Thank you for your attention!

Questions?