SWORN

Communication costs energy

Energy is finite

Protecting constrained nodes

Battery depletion attacks

Talk to a constrained node until its battery is gone

- Reception is expensive, even if information is discarded

Traditional protection: Firewalling

- Hard to make secure for UDP (source IP = password?)
- Puts a lot of application state into unrelated nodes

Giving control back to the device

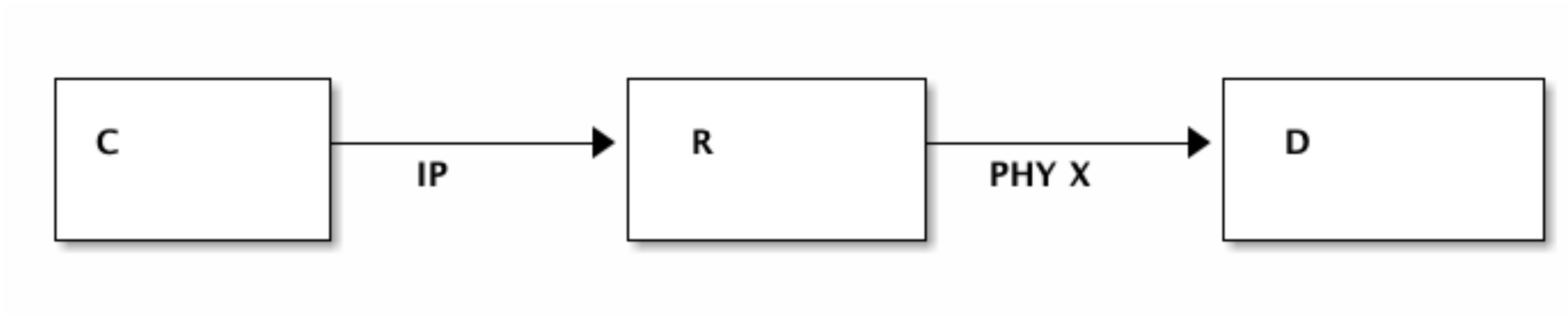
Device cannot protect itself

Energy expenditure is **authorized** by last hop router

Why not make the last hop router smart about that?

- Provide authorization to correspondent node
- Enable the last hop router to check it

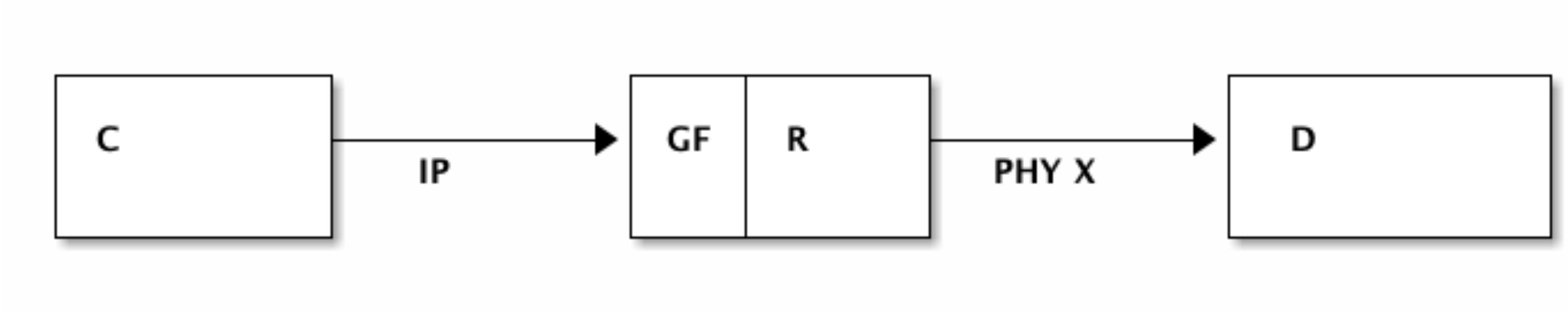
correspondent node C uses normal IP path to device D



last-hop router R talks to device D over PHY X

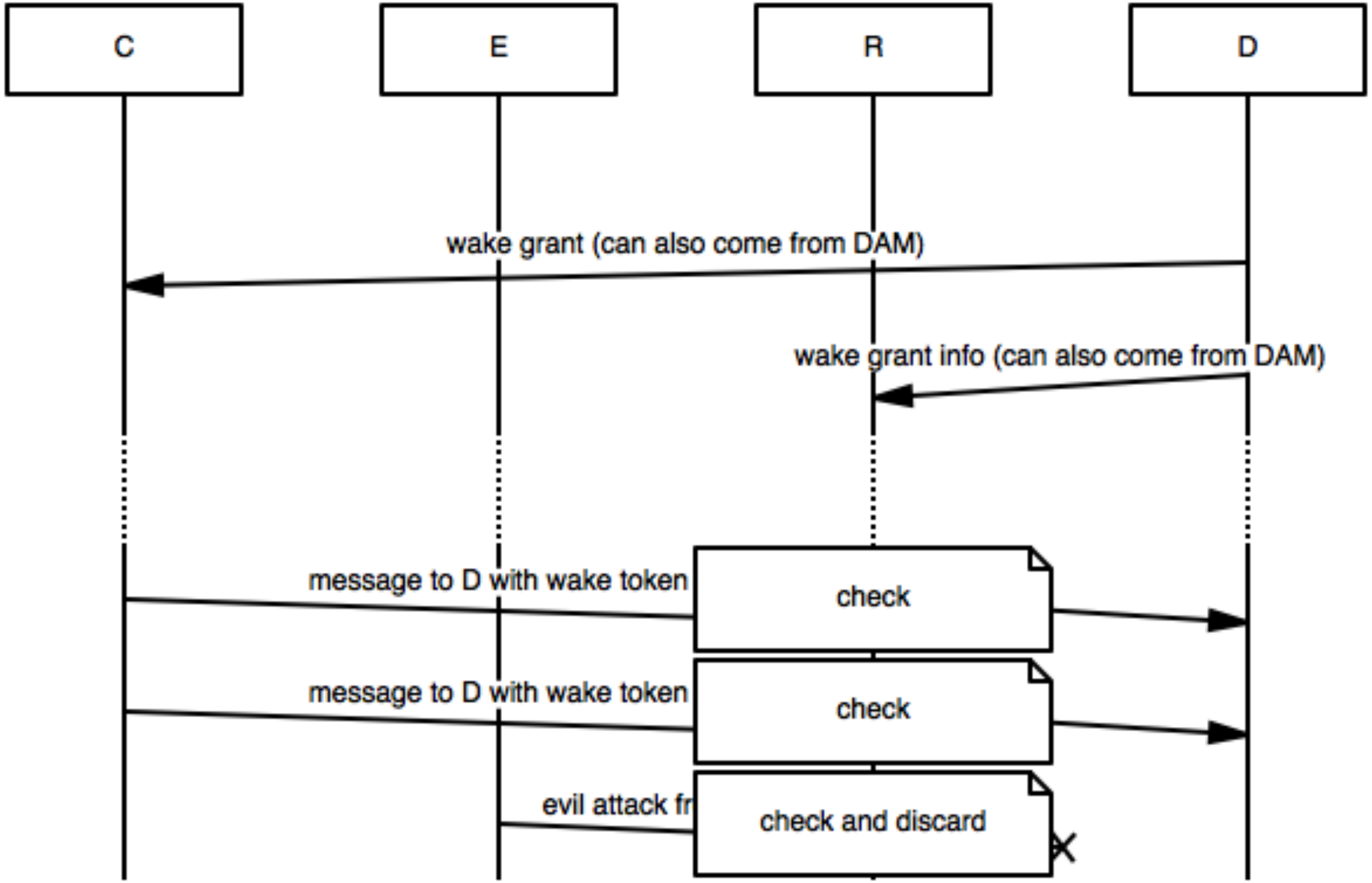
R can "wake" D over PHY X — expensive

before forwarding, R executes a gating function GF



C cannot bother (deplete the battery of) D without passing GF

How to make GF *secure* and *controllable by D*?


C
E
R
D
wake grant (can also come from DAM)
wake grant info (can also come from DAM)
message to D with wake token
check
message to D with wake token
check
evil attack fr
check and discard

Crypto: COSE-based

Wake grant (D ➔ C) is a CWE, packaging a grant key.

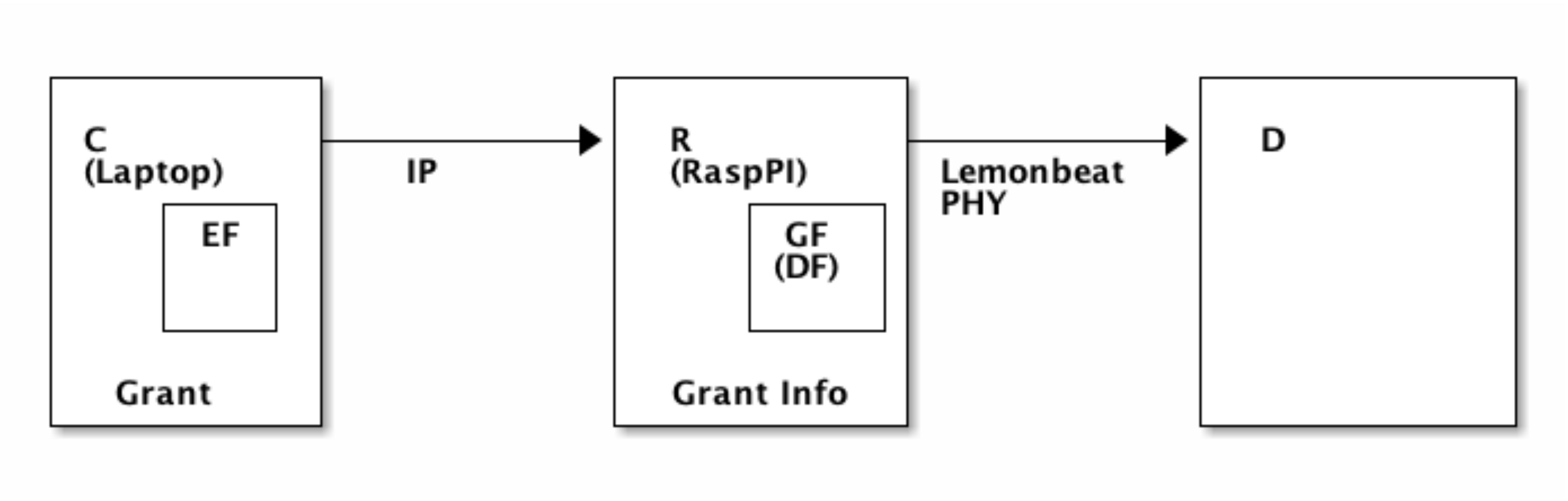
Wake grant info (D ➔ R) is same, possibly with a lifetime.

Wake token (C ➔ D, but really for R) is a CWS, containing `[serial: uint, wake-period: duration]`

Add some encapsulation magic, shake.

Demo Implementation @ IETF99 Hackathon

Basic SWORN scenario demonstrated between TZI and Lemonbeat GmbH:



Written up:

`draft-bormann-t2trg-sworn`

To do:

- Check the COSE structures that they really do what we want
- Check the "tunneling" (encapsulation) approaches

Acknowledgments

Thank you, Lemonbeat

Images

Wikimedia [^natteries], Makezine: [^arduinoserial]