QIFAN LU
2017.9.17

THE WISP EXTENDED RUNTIME

WHAT IS WISP?

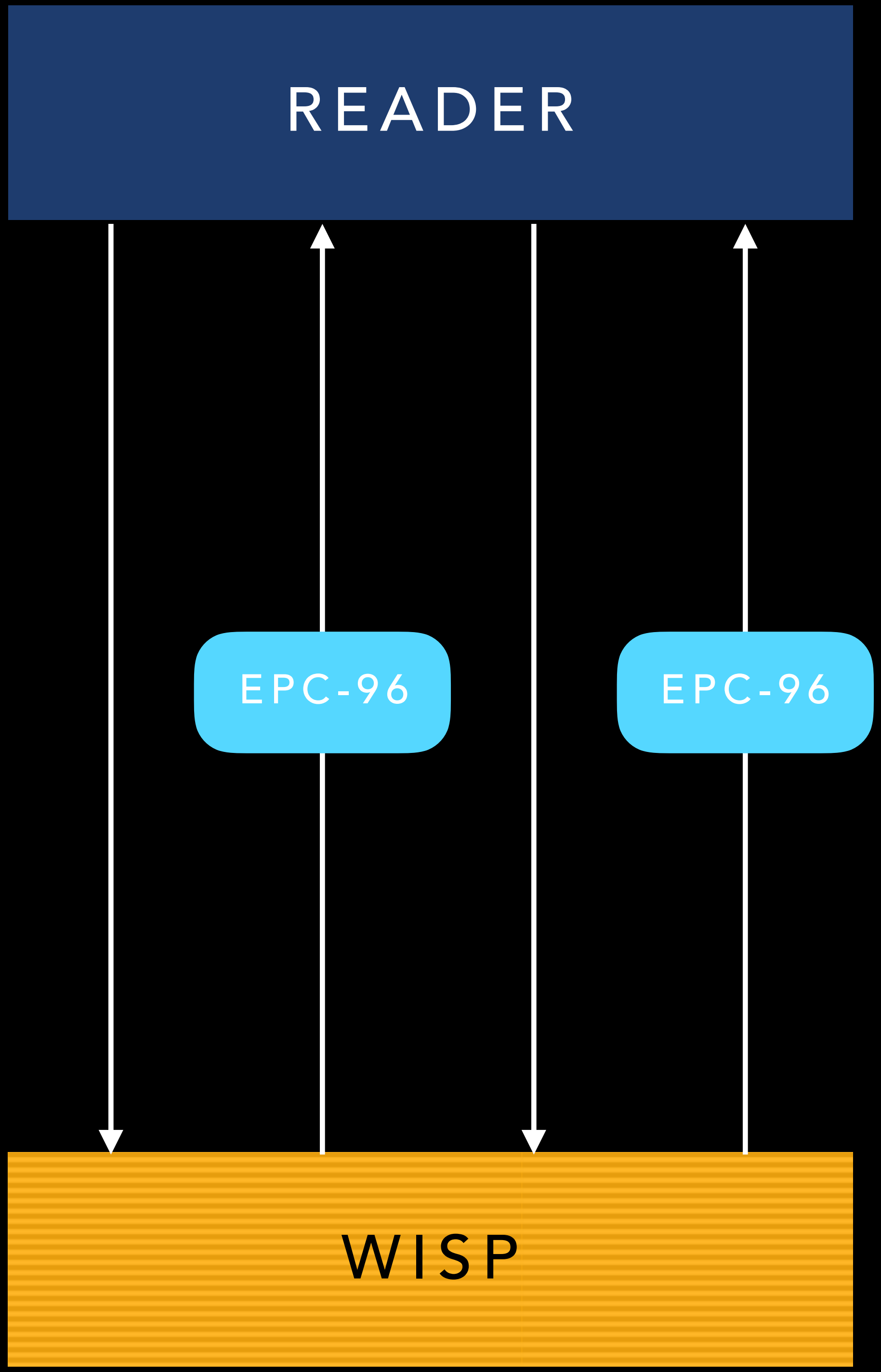
- Wireless Identification and Sensing Platform
- By UW Sensor System Lab
- WISP is a **Computational RFID**
  - Behaves like RFID (e.g. Powered wirelessly; complies to EPC C1G2 protocol)
  - Programmable (MSP430 micro-controller)
- The latest version is WISP 5.1

WHAT ARE YOU WORKING ON?

- Design a C interface for the WISP that can be used to manipulate remote files.
- Split into 3 parts as this is a complex task:
    - WISP Transmission Protocol (WTP)
    - Remote procedure call framework (u-RPC)
    - File operation and miscellaneous functions (WISP Extended Runtime)

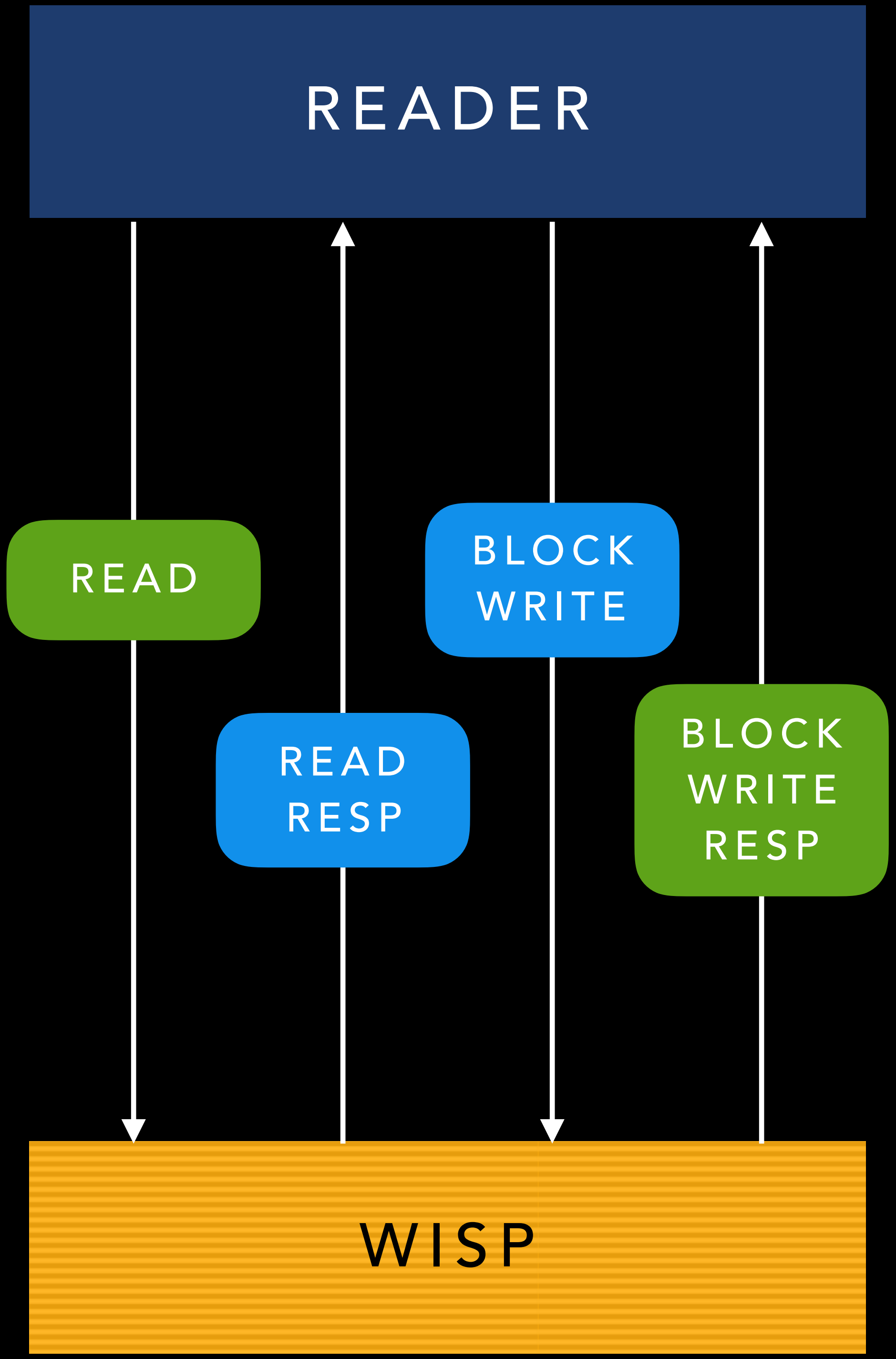
COMMUNICATE WITH WISP

- WISP to reader: EPC C1G2 Protocol
  - Reader periodically scans active RFID tags
  - Tags backscatter information about themselves
  - EPC-96 uniquely identifies a RFID tag
  - Reader commands: **Read**, **Write** and **BlockWrite**



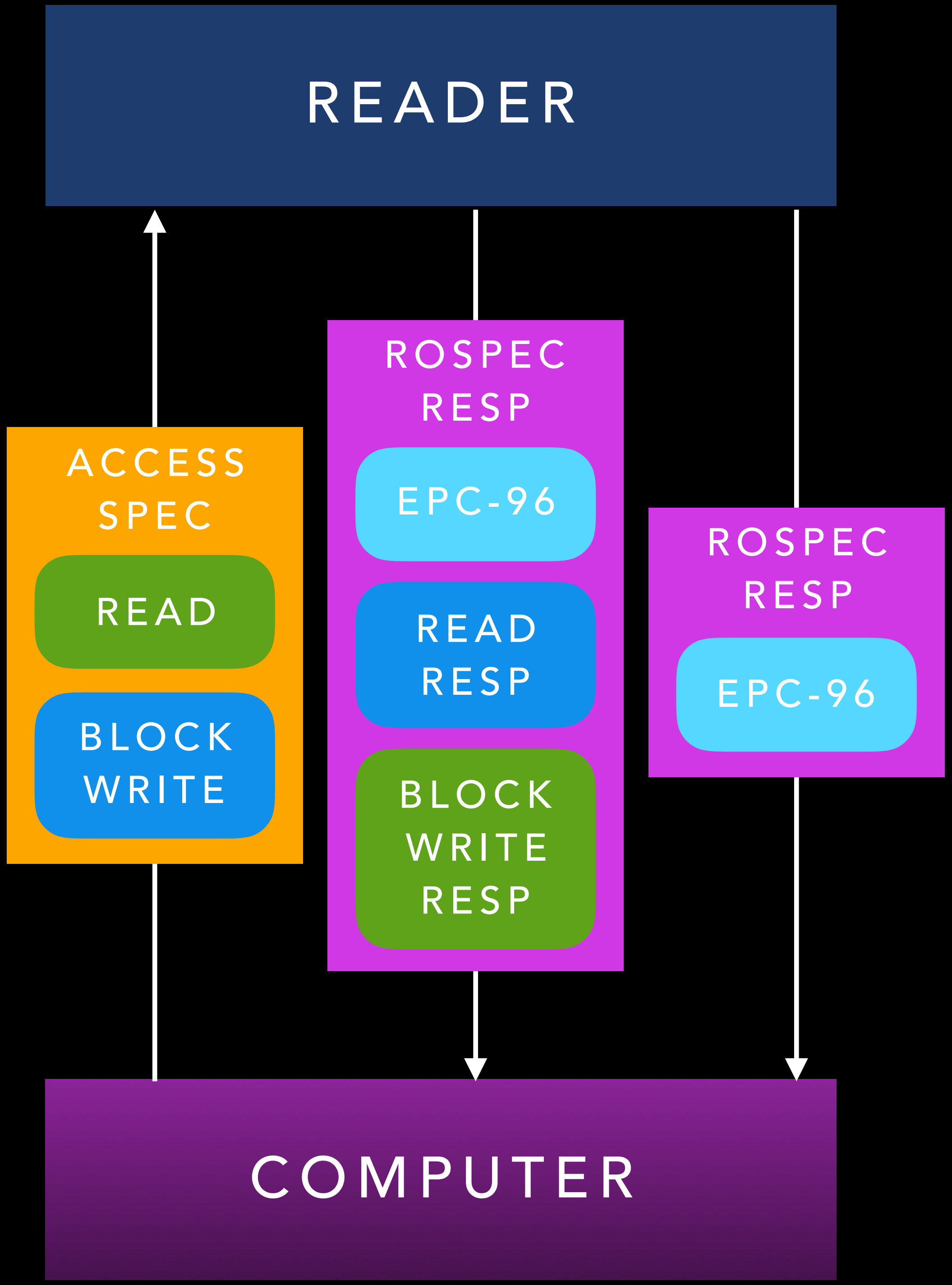
COMMUNICATE WITH WISP

- WISP to reader: EPC C1G2 Protocol
  - Reader periodically scans active RFID tags
  - Tags backscatter information about themselves
  - EPC-96 uniquely identifies a RFID tag
  - Reader commands: **Read**, **Write** and **BlockWrite**



COMMUNICATE WITH WISP

- Reader to computer: LLRP Protocol
    - **Read**, **Write** and **BlockWrite** operations represented by OpSpec messages
    - OpSpec messages further encapsulated inside AccessSpec messages, along with RFID target information and stop access condition
    - Computer receives tag reports and AccessSpec results in ROSpec_Response messages

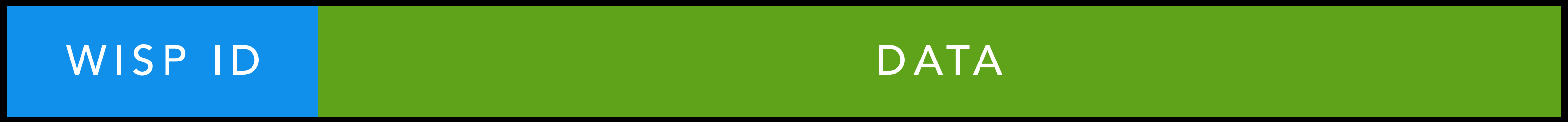


WISP TRANSMISSION PROTOCOL

- Problem: **Read**, **Write** and **BlockWrite** can fail because of poor signal or wireless interference.
- Solution: build a reliable transmission protocol on top of unreliable EPC commands.
- WTP borrows ideas from TCP, but is message-based and takes underlying EPC C1G2 protocol into consideration.

WISP TRANSMISSION PROTOCOL

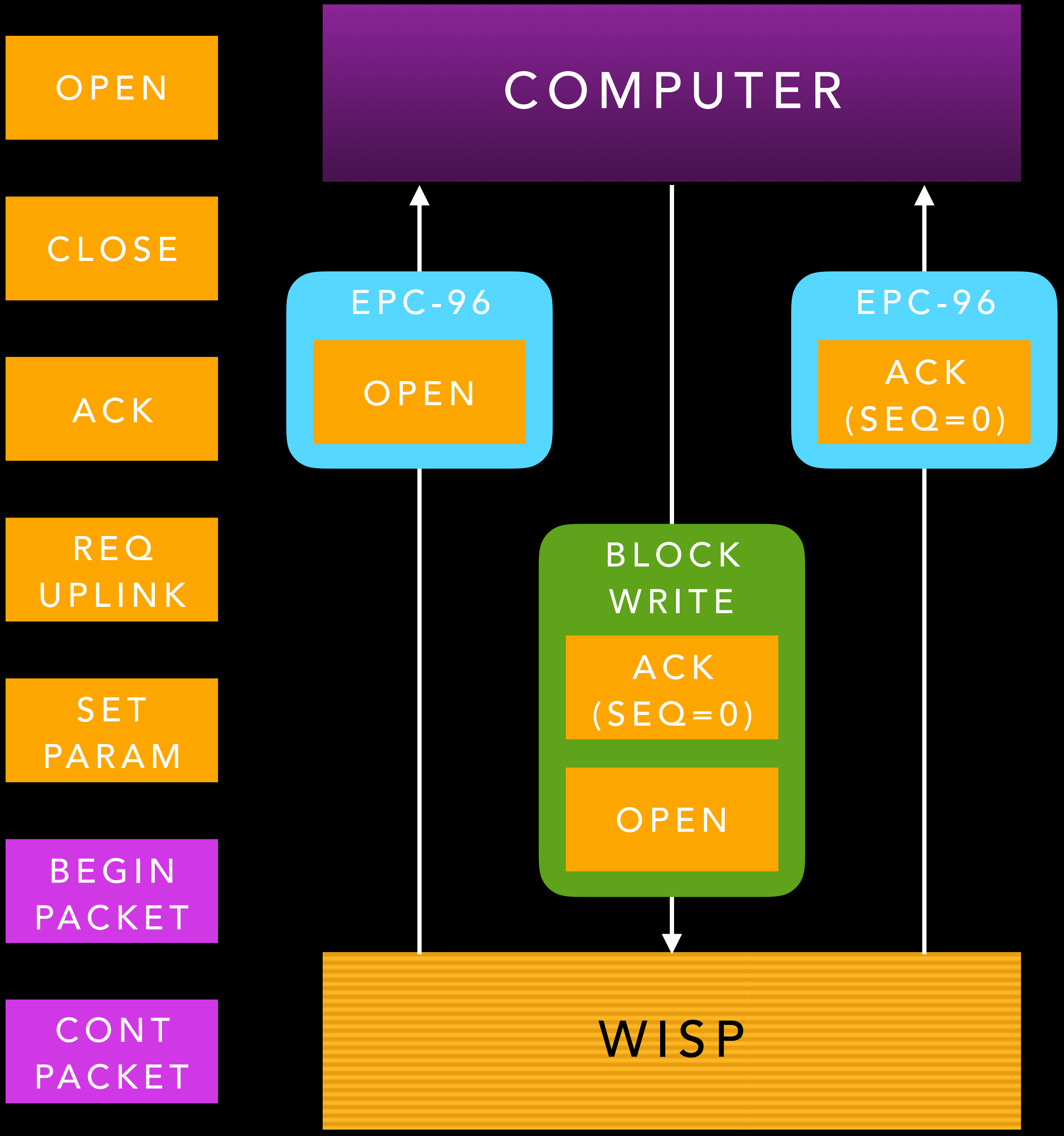
- How many ways to send data to the other side?
  - EPC-96 (Small; Uplink; By WISP)
    - 16-bit WISP ID is sufficient; remaining 10 bytes can be used to send data to reader



  - Read (Large; Uplink; By computer)
  - BlockWrite (Large; Downlink; By computer)

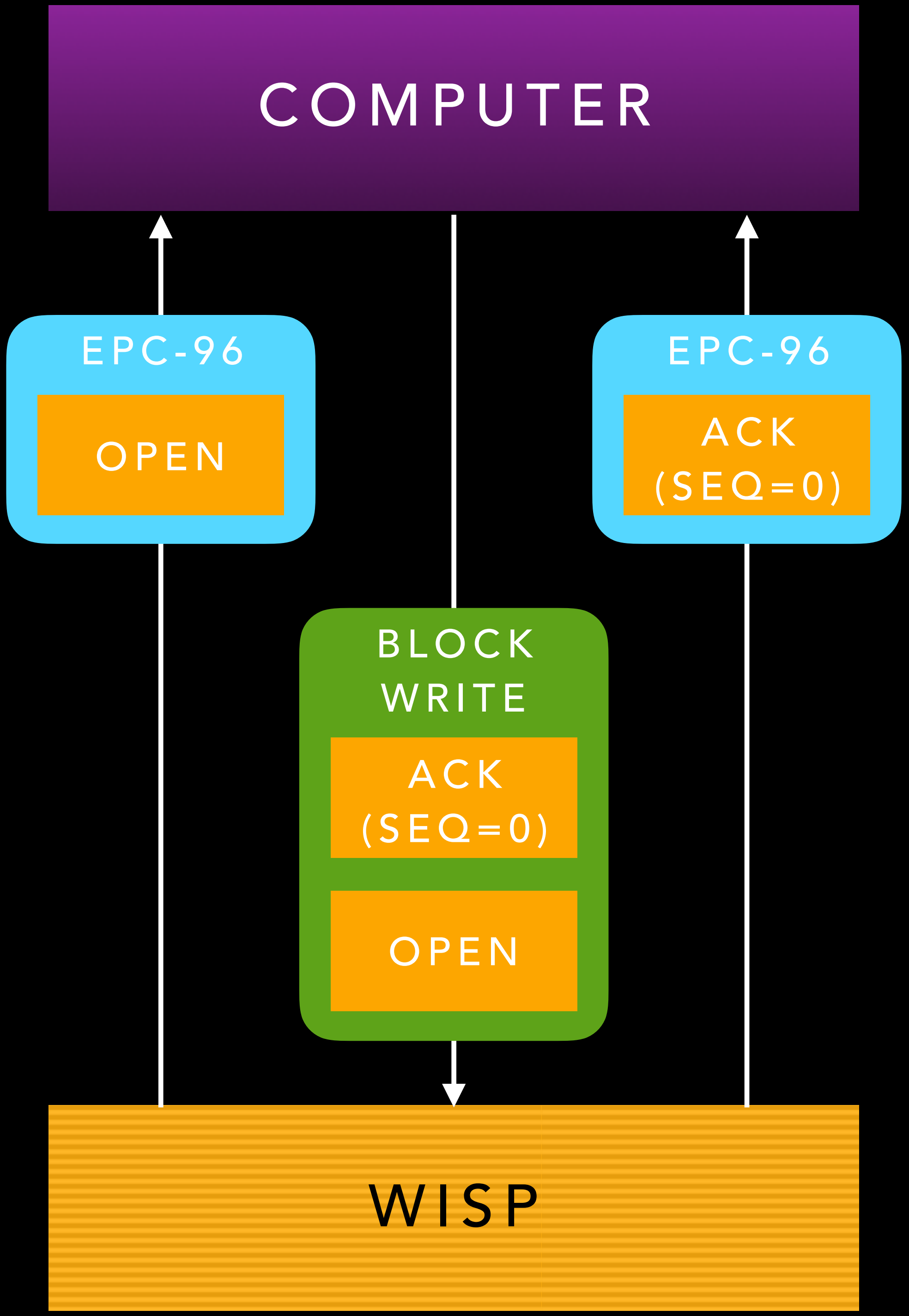
WISP TRANSMISSION PROTOCOL

- WTP packets
  - Control packets
  - Data packets
- An EPC-96, Read or BlockWrite may contain multiple WTP packets to increase efficiency.



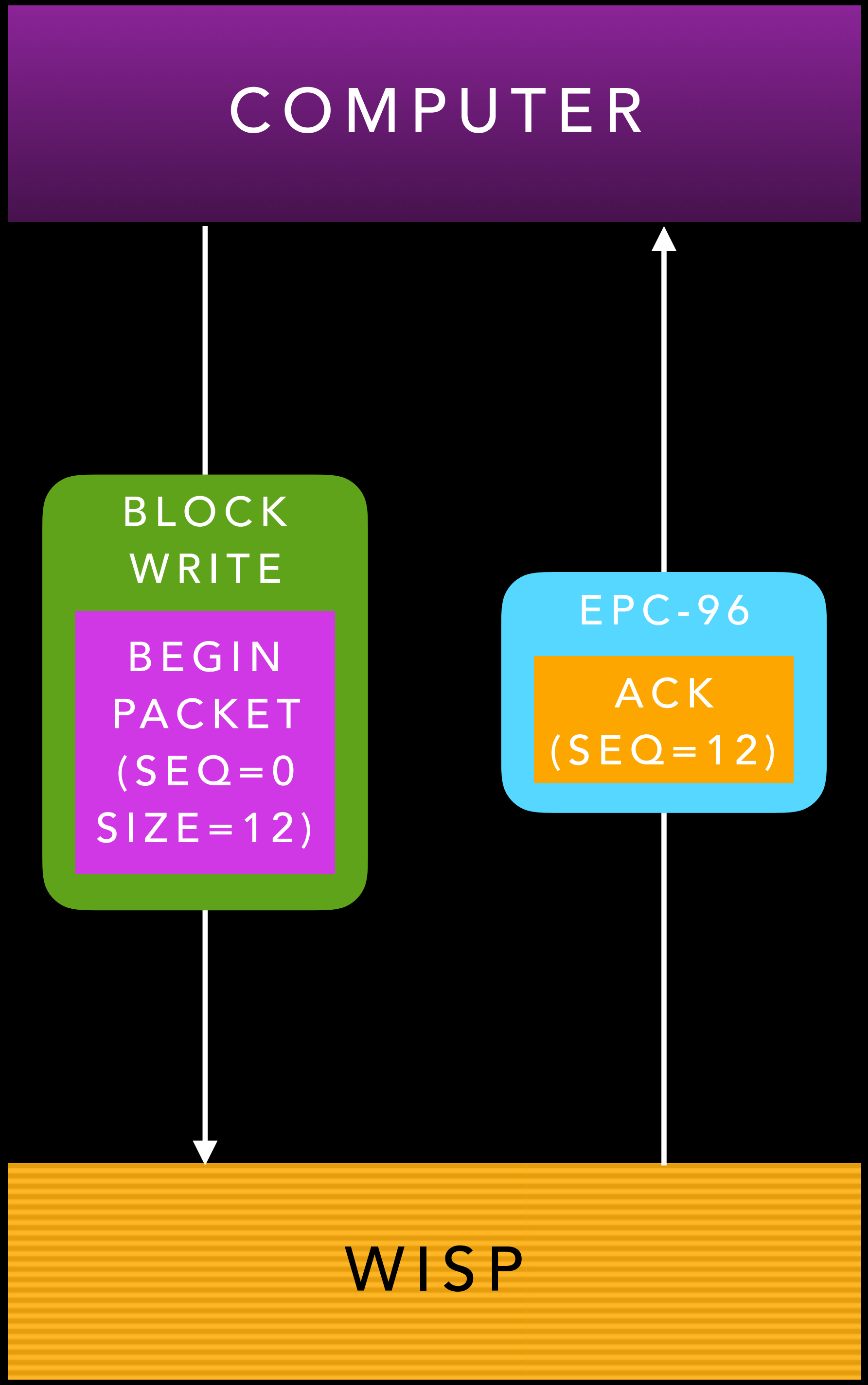
WISP TRANSMISSION PROTOCOL

- WTP uplink and downlink are opened separately (Similar to TCP handshake)
- For downlink, messages are fragmented and the fragments are sent to the other side with sequence number and fragment size.



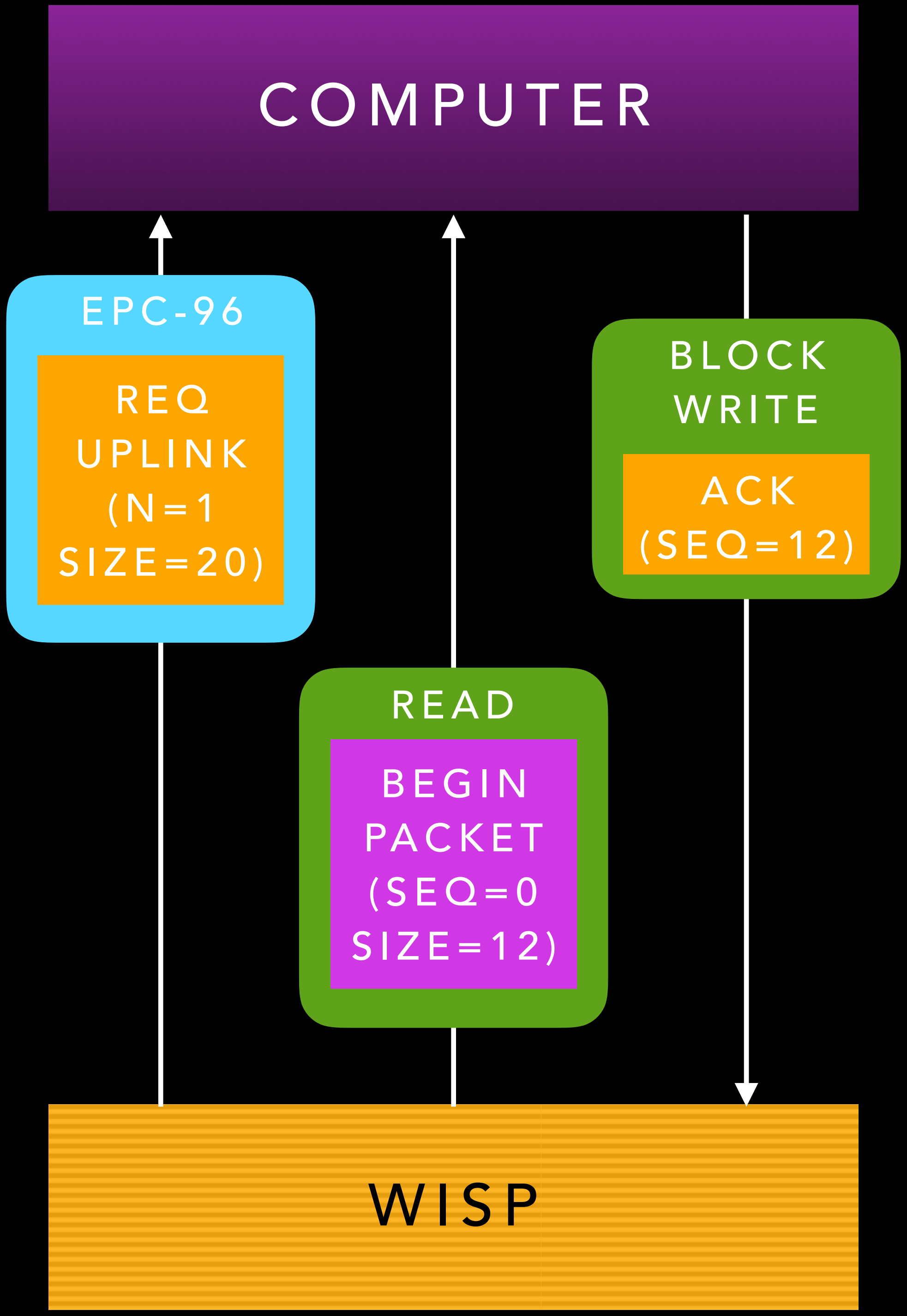
WISP TRANSMISSION PROTOCOL

- The receiver sends an acknowledgement packet when it receives and validates the fragment.
- If the sender of the message does not receive corresponding acknowledge in time, the fragment is then retransmitted.



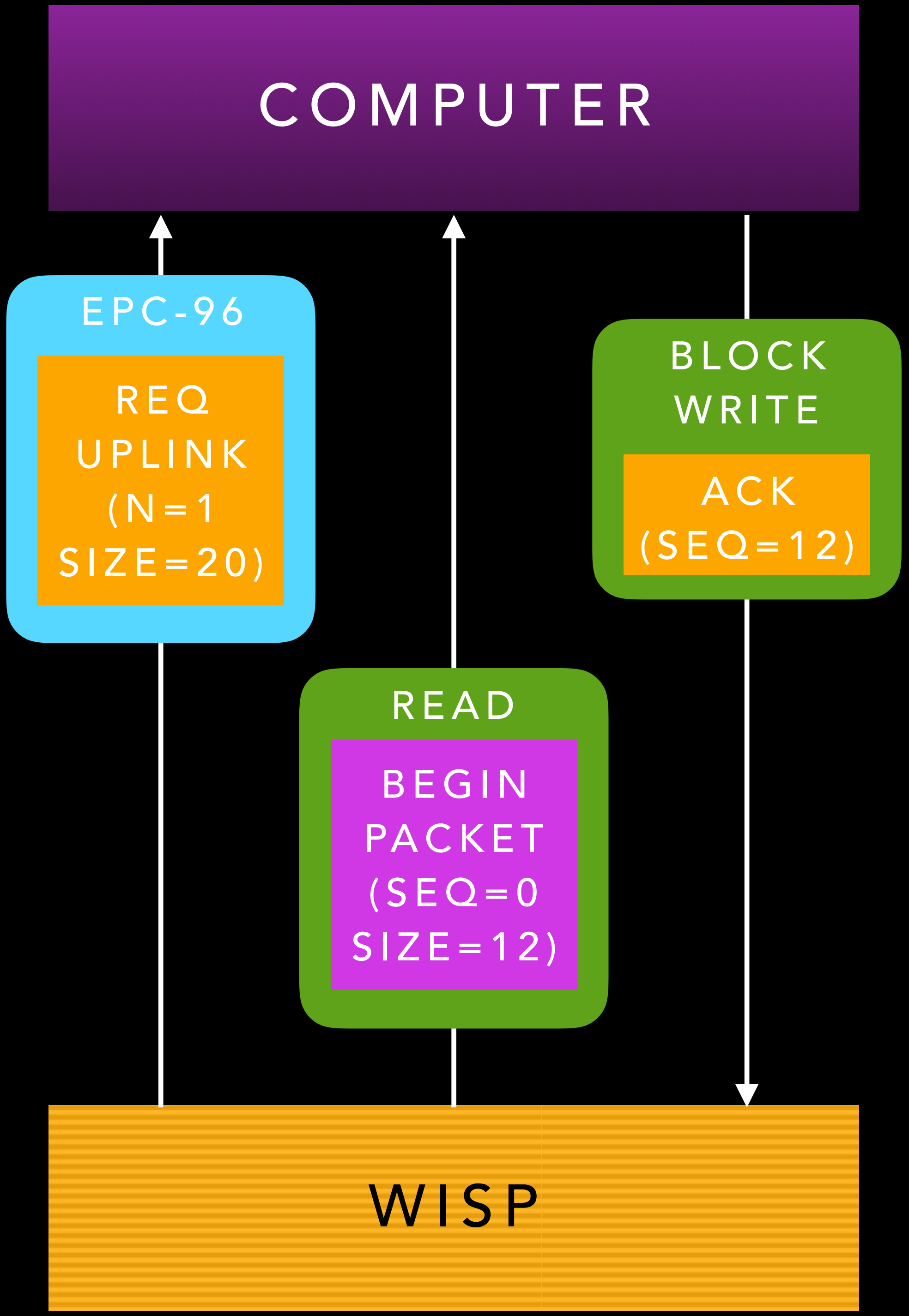
WISP TRANSMISSION PROTOCOL

- Uplink transmission happens in a similar manner, and data gets sent through Read instead of EPC-96.
- Since Read can't be initiated by WISP, we need to work around the problem by "requesting uplink" from the computer side.



WISP TRANSMISSION PROTOCOL

- The WISP sends a "requesting uplink" packet containing the count and the sizes of the Read.
- The computer reads the WISP on behalf of it and the WISP loads the data to send after each Read operation.



WISP TRANSMISSION PROTOCOL

- The WTP server-side program monitors the status and the number of words written in a Read or BlockWrite.
- When Read or BlockWrite failed, the server decreases the maximum size of Read or BlockWrite OpSpec by 2, causing the connection to be throttled.
- When Read or BlockWrite succeeds and the size of data sent is the maximum allowed size, the maximum allowed size is increased by 2.

U-RPC

- A remote procedure call (RPC) framework built for embedded devices.
- u-RPC hides the serialization and deserialization process in the framework, so you can simply add functions to one endpoint and call them from the other.
- To call remote functions:
  - Query remote function handle by name (Optional)
  - Call remote function by handle

WISP EXTENDED RUNTIME

- The WISP Extended Runtime (WISP ERT) extends WISP with remote functionalities on computer or cloud services.
- Server-side API
  - Pluggable services consisting of a group of functions and constants
  - Constants synchronized to WISP at initialization and functions added to u-RPC endpoint
  - The filesystem demo service (“fs”)

WISP EXTENDED RUNTIME

- Client-side API
  - All initialization stuff (initialize data structures; sets up RFID loop; connect to WTP server; etc)
  - Provides constants and wrapper functions for server-side file operation routines.
  - Context switching and sackful coroutine support: more pretty asynchronous code and non-volatile stack memory.

REPOSITORY & DOCUMENTS

- u-RPC
  - Repo: https://github.com/lqf96/u-rpc
  - Wiki: https://github.com/lqf96/u-rpc/wiki
  - C API: https://lqf96.github.io/u-rpc/c/html/index.html
  - Python API: https://lqf96.github.io/u-rpc/python/html/index.html

REPOSITORY & DOCUMENTS

- WISP Extended Runtime
  - Includes WTP and WISP ERT source code
  - Repo: https://github.com/lqf96/wisp-ert
  - Wiki: https://github.com/lqf96/wisp-ert/wiki
  - Client API: https://lqf96.github.io/wisp-ert/client/html/index.html
  - Server API: https://lqf96.github.io/wisp-ert/server/html/index.html

QUESTIONS?

THANKS!