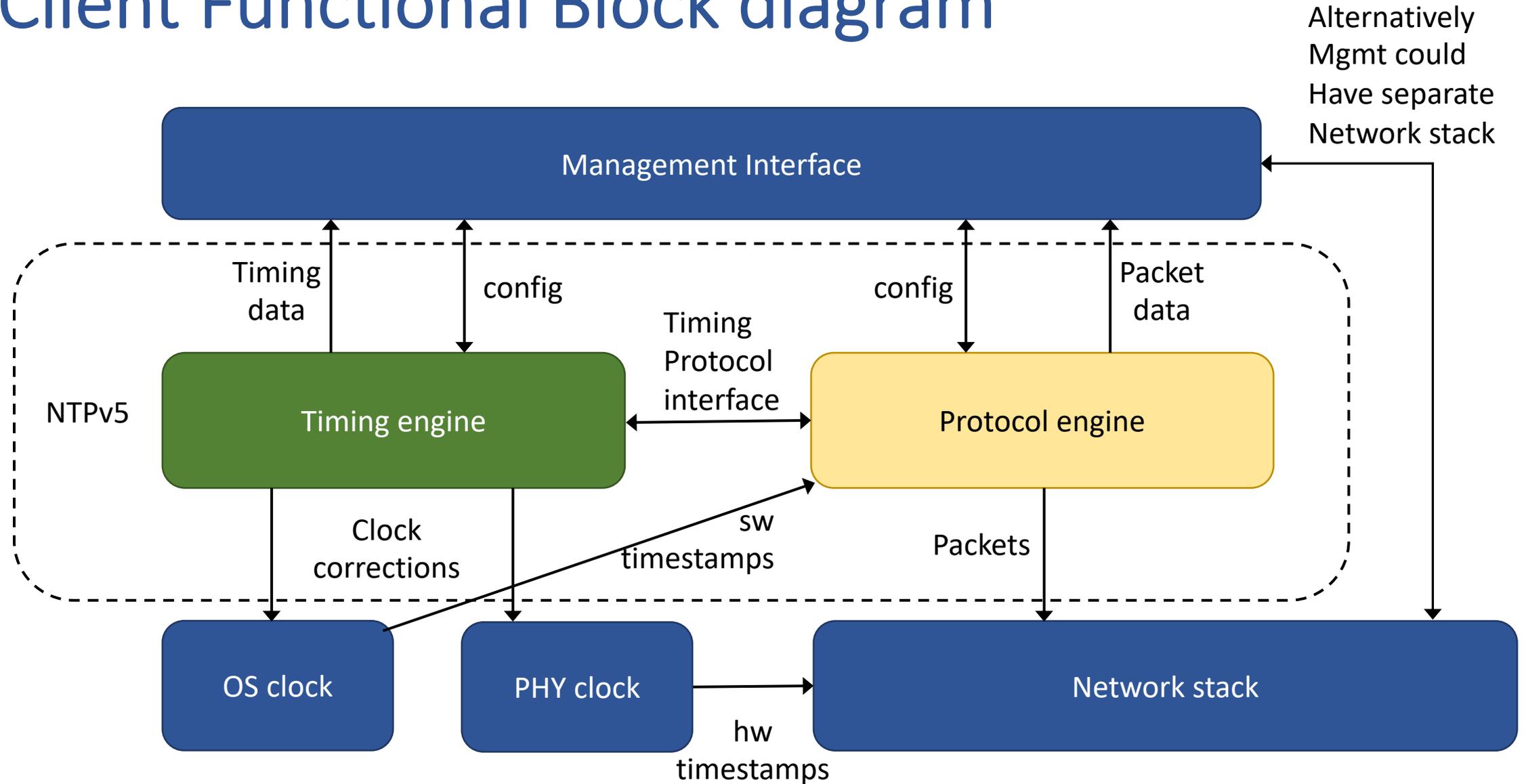


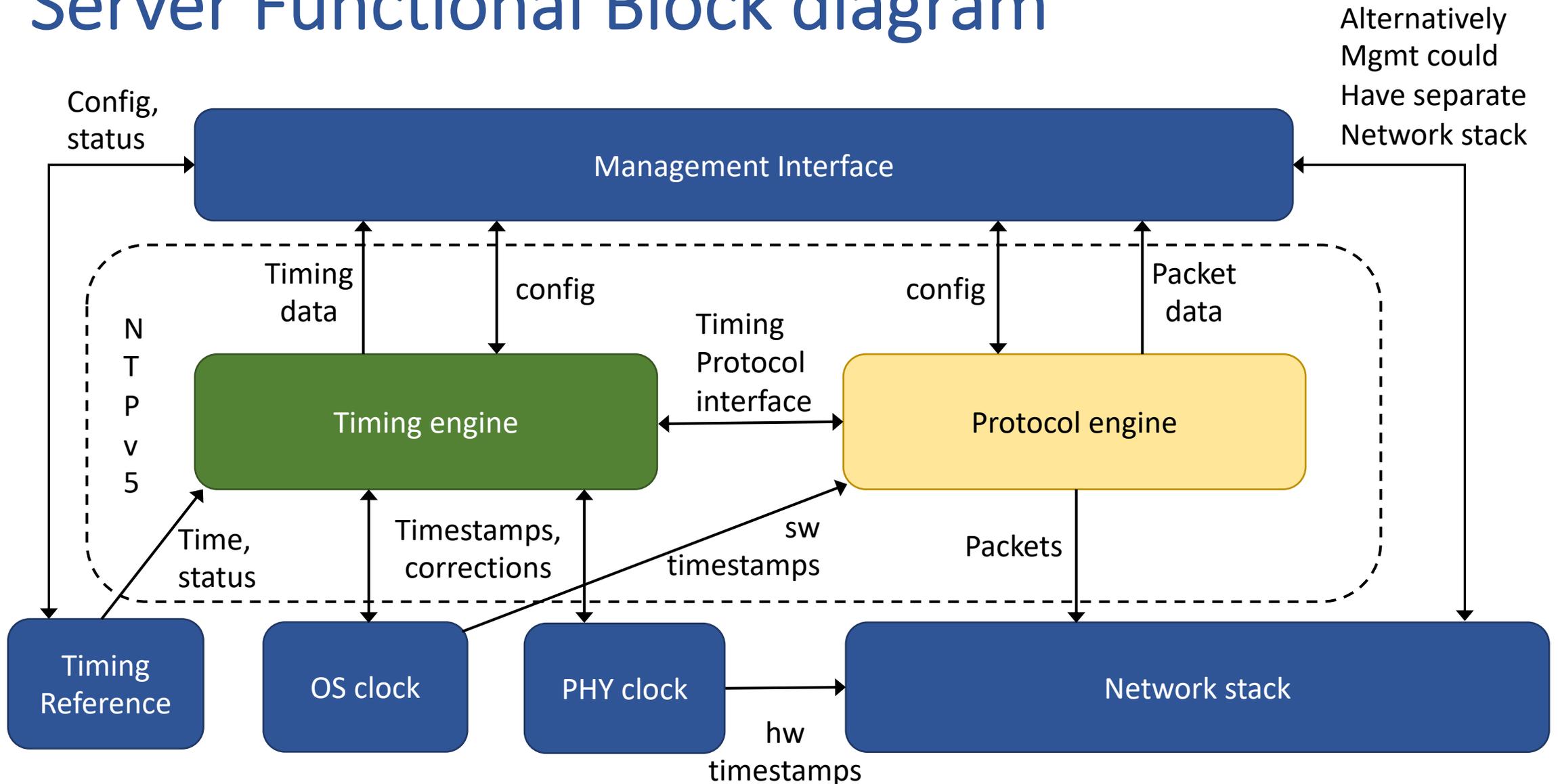
NTPv5 Modular Architecture

- Proposal: Define NTPv5 as two interacting subsystems
 - Timing engine
 - protocol engine
- Purpose: allow different timing engines to be defined for different applications
 - For example, General purpose IT (time for logfiles, security ticket times outs, ...)
 - For example, precision timing for financial networks
- Purpose: allow different protocol engines to be defined for different applications
 - For example, with and without security

Client Functional Block diagram



Server Functional Block diagram



Functional Block Diagram Notes

- Local clocks
 - OS system clock (SW timestamps)
 - Steerable counter on a PHY chip (HW timestamps)
 - Custom HW clock. Often implemented in time servers or cyber physical systems
- Timing engine
 - Clients do not need to read local clock, only to determine its offset via NTP
 - Server needs to read local clocks to steer them to the timing reference
 - Timing reference: GNSS receiver, PTP input, 1PPS input, etc

Protocol Engine

- Interfaces with network stack
 - Builds packets for transmission
 - Software layer timestamps
 - Parses packets upon receipt
- Executes network security
- Determines when to send packets
 - Based on average packet rate from timing engine
- Passes received information to timing engine
 - Timestamps and timing metadata
 - Message status, such as expected message not received, security working, etc

Timing engine

- Selects servers to receive time from
 - Allows for optimization based on analysis of timing data
- Analyze received timing information
 - Outlier detection and removal
 - False ticker identification and removal
 - Lucky packet pre-filters
 - Generate timing statistics
- Clock control
 - PLL filter
 - Clock corrections
- Report statistics to management interface

Timing engine - protocol engine interface

From timing engine

- List of target server IP addresses
- Average packet time interval for each server

From protocol engine

- Server not responding flags
- Received packet data
 - Four timestamps
 - Root delay & root dispersion
 - Reference ID
 - Leap second flag
 - Server stratum
 - Security on/off flag

Feedback welcome

- Thanks to Ulrich Windl, and Hal Murry for astute questions and suggestions
- Looking for coauthors for a draft architecture document
- doug.Arnold@meinberg-usa.com