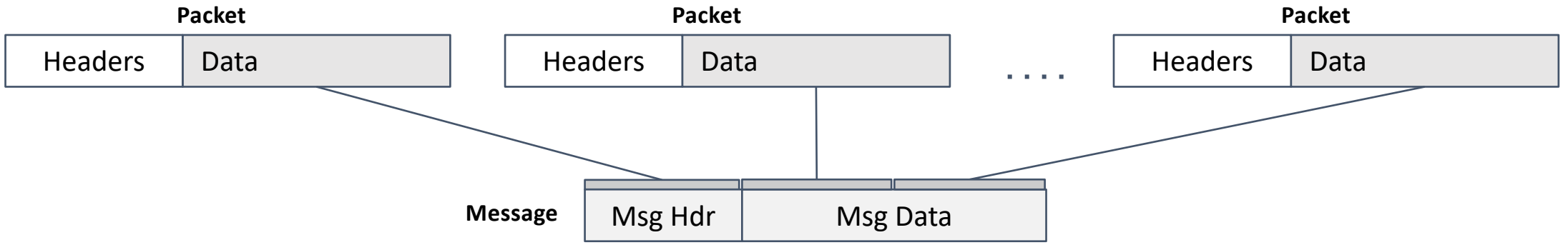
P4 Evolution – From Packet Processing to Message Processing

Vipin Jain
Co-Founder & CTO
Pensando.io

PENSANDO

Message Processing



Packet Processing	Message Processing
Parse, Match, and Actions on Packets	Parse, Match, and Actions on Messages
Typically upto L4 (TCP/UDP + Encap), or within Packet	Beyond Layer4 (session, applications)
Almost Always Stateless	Almost Always Stateful
Changes Less Often (e.g. new encaps)	Frequently Changing (new apps often)
Applicability: Host or Network Devices	Applicability: Commonly Host (closer to applications)

So, what's common? It is P4

PENSANDO

Use Cases for Message Processing

- Storage Virtualization
  - NVMe Over TCP
- Service-Mesh (Istio, e.g.) Datapath
  - API Routing, Load Balancing, API Authentication, Metrics, SPIFFE, TLS Security
- App Level Security
  - Web Application Firewall (WAF), NextGen Firewall (NGFW), Signature Parsing
- 5G Edge Computing
  - Firewall/DPI (along with UPF)
- Virtual Network Functions Acceleration
  - SSL Gateways, Firewall (IDS, Anomaly Scans), CDN, CPE (BFD), etc.
- Application Troubleshooting and Observability
  - Message Tracing, Performance Monitoring, Application Level Latency

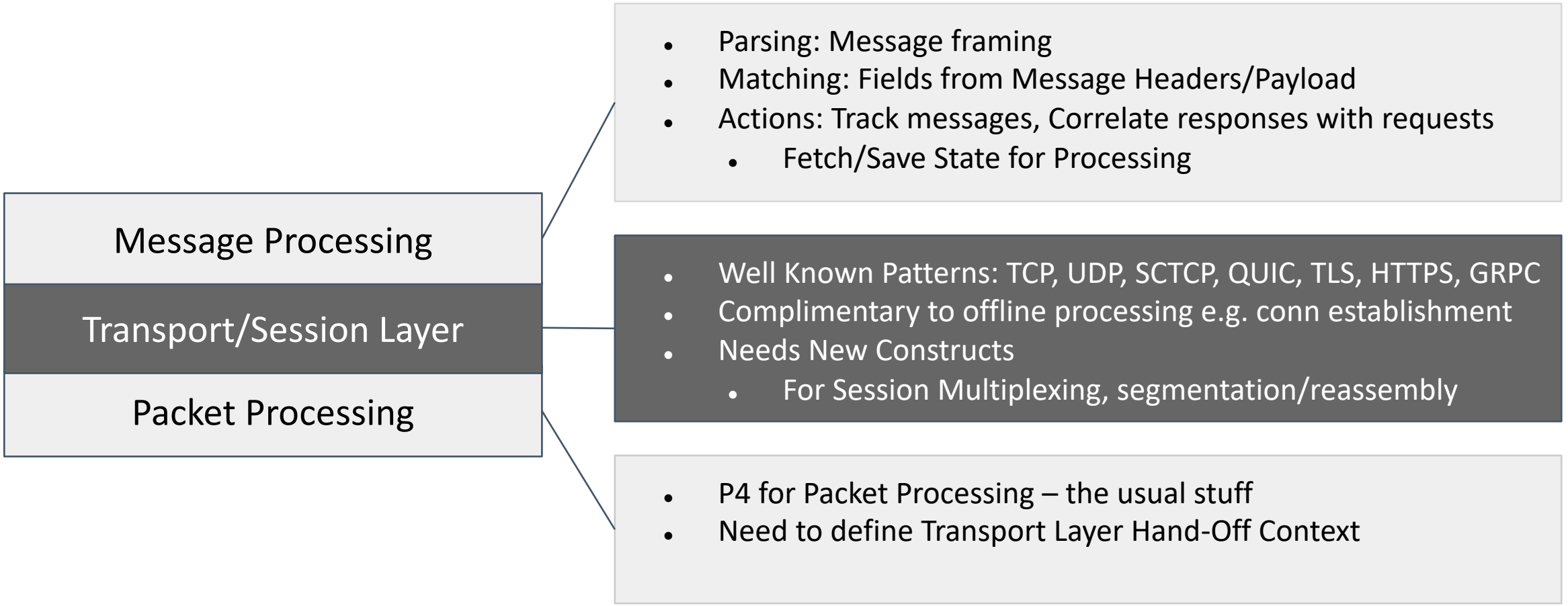
PENSANDO

Implementation Challenges

- Transport and Session Layer Implementation
  - Multiple Implementations: TCP, SCTCP, UDP, HTTP, GRPC, etc.
  - Typically best done as general purpose software
- Complex, Stateful FSMs
  - Both transport and application state-machines are complex
  - Need multi-state transition with persistent metadata for various states
- Encrypted Data
  - Parsing session or application information is not possible
- Higher Scale
  - Usually over one connection, multiple message states e.g. iSCSI commands
- Heavy on Memory Operations
  - Load/Store state at high scale, need DMA friendliness

PENSANDO

Architectural Layering



Transport/Session Layer Implementation

- Well Known Patterns: TCP, SCTCP, QUIC, TLS, HTTPS, GRPC
- Hand-Off from Packet Processing Layer
  - Context Handoff
- Common Transport Functions: Segmentation and Reassembly
  - Timers: Retransmission, Background Threads
  - Header Compression/Decompression
  - Buffering: Memory area to hold the messages for long period
  - Transport/Session control state: Maintain per flow transport state
  - Packet Generation/Termination: Originating Acks or Terminating Ack-only Packets
- Complimentary to offline (non P4 e.g. CPU) processing
  - Connection/Session establishment (typically one time)
- Session Multiplexing (e.g. GRPC, TLS, HTTPS)
  - Multiplexing multiple contexts over one TCP connection
  - Somewhat akin to message processing (discussed in the next slide)

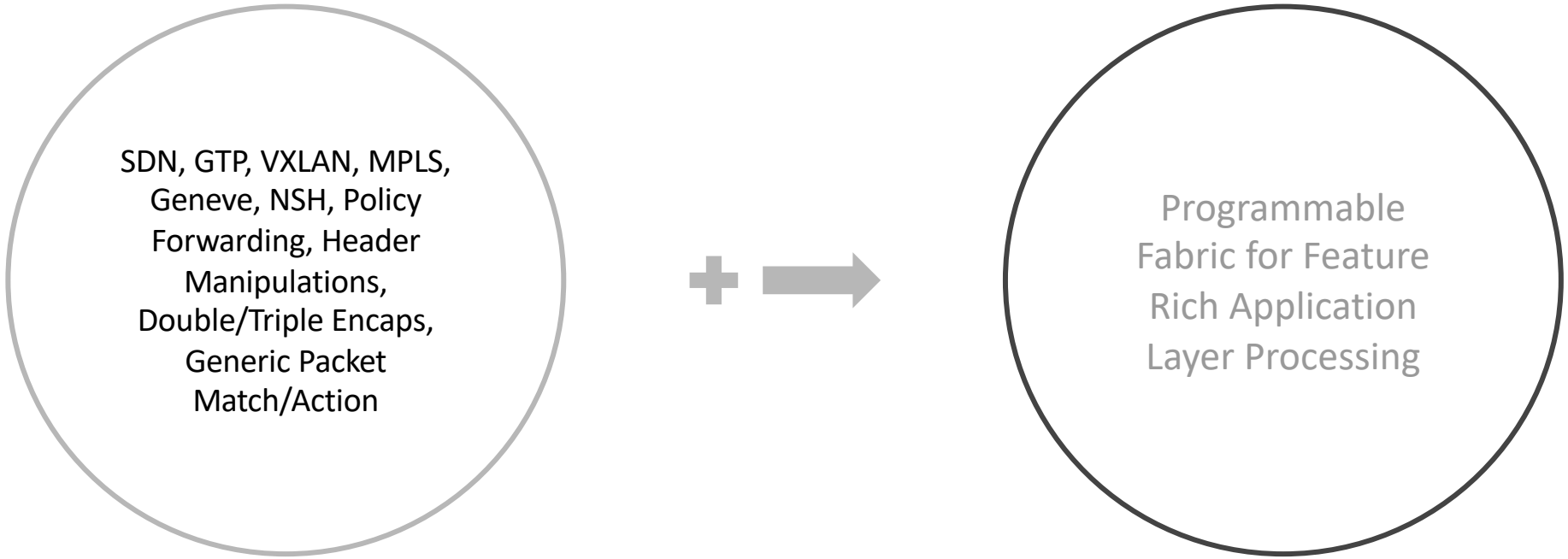
PENSANDO

Message Processing

- Message can leverage any transport
- Message Parsing
  - Typically at the beginning of the message, can be expressed in P4
  - But where does it begin?
    - Message length determination
    - Not always present in the message header e.g. Redis line protocol
  - Message Contents: Sometimes binary format, but can be text oriented e.g. http
- Match Criteria: Important Message Header Fields
  - Verbs: commands or opcodes e.g. iSCSI read vs. write operation
  - Identifier: Message key e.g. URL in HTTP, topic in message bus, or key in KVStore
    - For request-response correlation
  - Response: Status code, response data
- Message Actions
  - Per Message State (keyed by message ID)
    - Track: Metrics, Opcodes, Result, Latency, Backend Selection (Routing), etc
  - Actions: Drop, Route, Forward, Modify, Record, Analyze
  - Generate Packets/Messages e.g. IPFIX records, beacon packets

PENSANDO

Let us evolve P4 together



Packet Processing

Message Processing

Thank You

Email: jainvipin@pensando.io
LinkedIn: https://www.linkedin.com/in/jain-vipin/
Twitter: https://twitter.com/TheVipinJain
Pensando: https://pensando.io

PENSANDO