

HTTP/1 Protocol

Presenter: Hans Wang, AI Research Scientist, SPR lab, Intel Labs.

Contributor: Gordon King, Cloud Software Architect, SST.

Tuesday, April 26, 2022



Purposes

- Share an idea for an L7 protocol, HTTPPA/1, to establish trusted communication channel between HTTP endpoints using remote attestation.
- Invite feedback, contribution or collaboration.
- Notice: HTTPPA/2 is still in legal review.

Outline

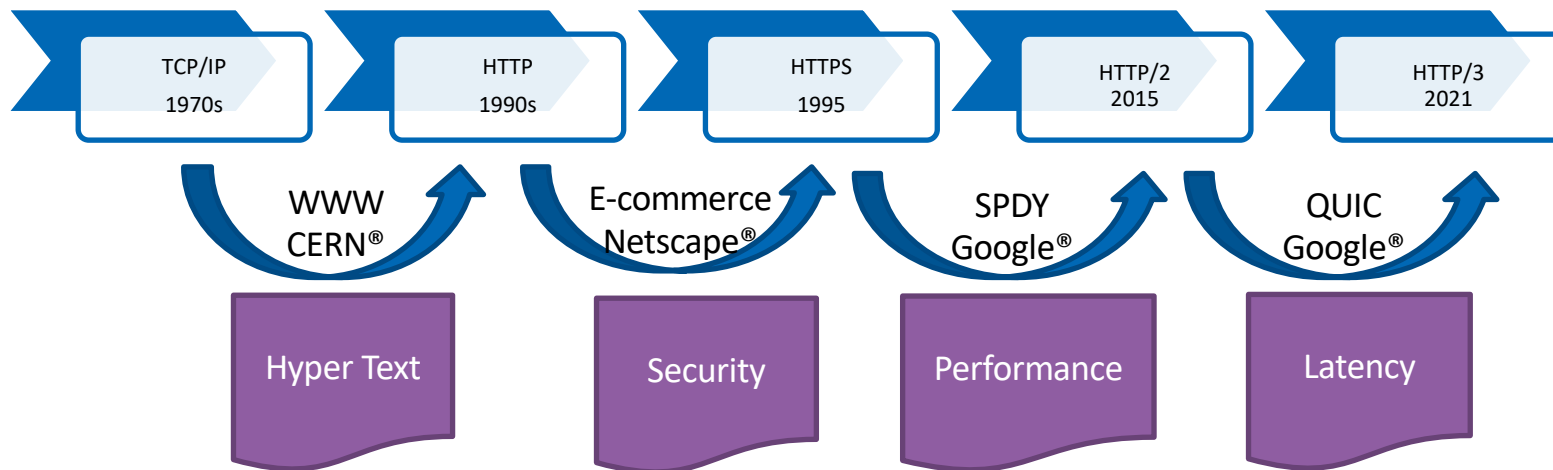
- Motivation
- Background
- Problem statements
- HTTPA
- Summary

Motivation

- Trusted federated learning requires building trust with remote parties.
 - How can we trust or attest the remote AI service?
- Customer's feedback to request a standard way to use attestation with better usability.
- Multiple confidential computing platforms need unification for communication protocol.

Background

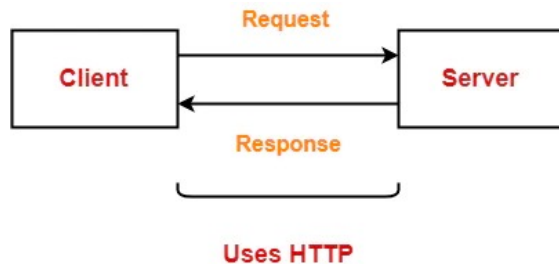
Internet Protocol Evolution



Background

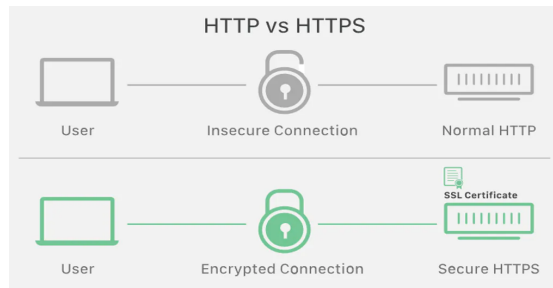
■ HTTP

- A protocol to define communication of request/response between a client and a server.



■ HTTPS

- HTTP over TLS/SSL
- **Not** a separate protocol from HTTP.
- TLS/SSL handshake to establish a secure connection:
 - TLS/SSL certificate for server authentication
 - TLS/SSL encryption for confidential communication
 - TLS/SSL message digest for authentic data integrity



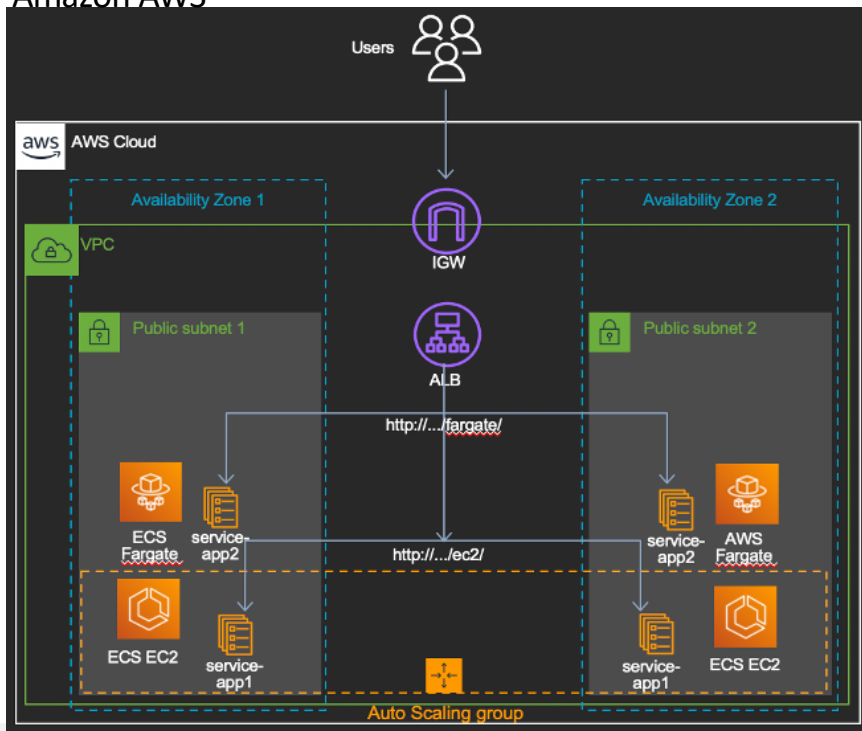
Problem statements

Problem statements

- Assumption: TEE-aware services will be popular in the future.
- Context:
 - A website can host multiple web services, many of them relying on third-party software vendors.
 - Webservices are vulnerable to the host, and vice versa.
 - Increasing demands for **microservices**, and **L7 load balancers**, and etc.
- **In most HTTPS communication, TLS terminates at the application gateway or L7 load balancer. This imposes security concerns of vulnerabilities, leading to lack of confidence in trustworthiness.**
- **Modern web services are lacks of trust.**
- **What's current solutions?**
 - **HTTPS** alone providing **secure channels** is not enough protecting web services for building trust.
 - Existing remote attestation mechanism adds high complexity for development and requires high learning curve for non-security developers, e.g., AI experts, data experts, software engineers, and etc.
 - Big players are proposing their own solutions to solve this problem.
- **We need a standardized approach by simply extending existing protocol to build trust for using web services: HTTPA.**
 - HTTPA aims to establish trusted end-to-end communication with the attested end point.

Industrial URL Routing and Load Balancing (LB)

Amazon AWS

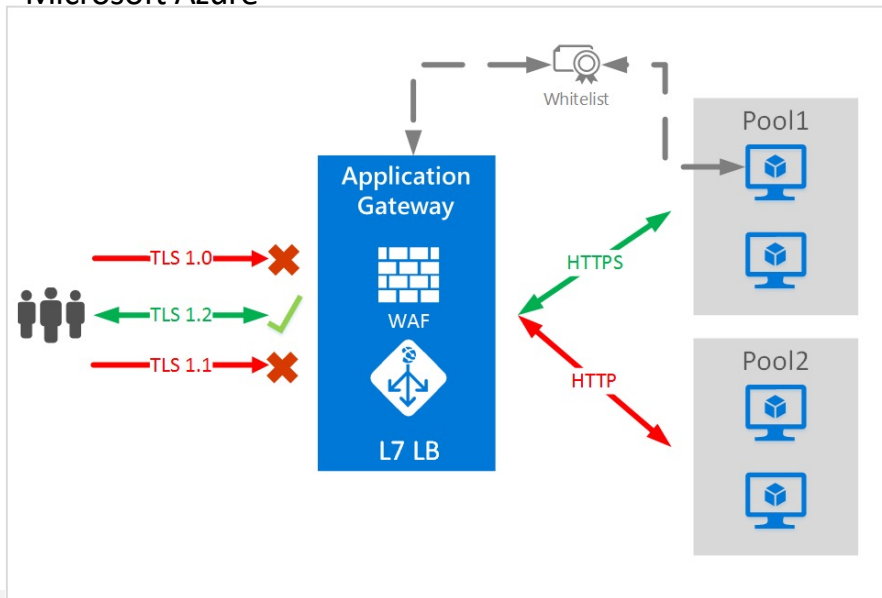


<https://aws.amazon.com/blogs/containers/using-aws-application-load-balancer-path-based-routing-to-combine-amazon-ecs-launch-types/>
<https://docs.microsoft.com/en-us/azure/application-gateway/ssl-overview>
 Intel Labs Security & Privacy Research

“Application Gateway supports **TLS termination at the gateway**, after which traffic typically **flows unencrypted** to the backend servers” – Microsoft Azure

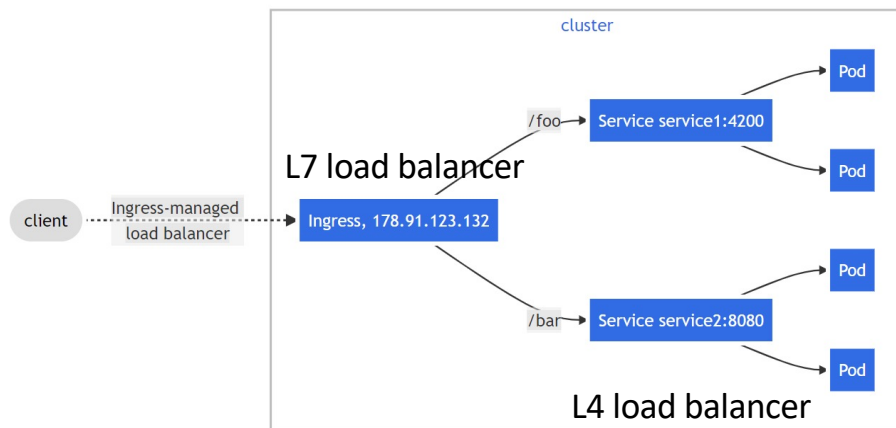
“Application Gateway then **initiates a new TLS connection** to the backend server and **re-encrypts data** using the backend server's public key certificate before transmitting the request to the backend.” – Microsoft Azure

Microsoft Azure



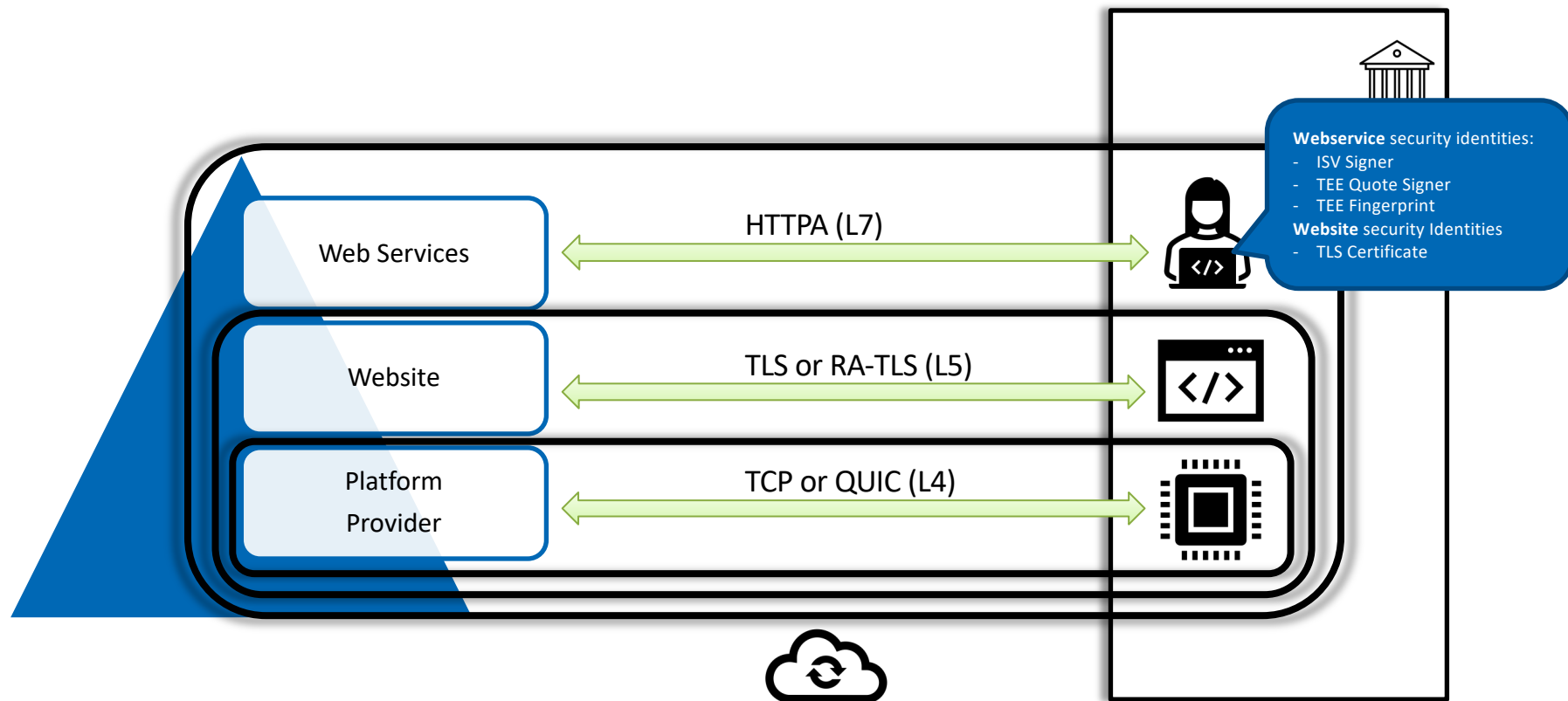
Kubernetes Ingress

- “The Ingress resource only supports a single TLS port, 443, and assumes **TLS termination** at the ingress point (traffic to the Service and its Pods is in plaintext).”



HTTPA/1

HTTPPA Stack for Internet



What is HTTP?

Both HTTP and HTTPA reuse original untouched TLS to secure traffics.

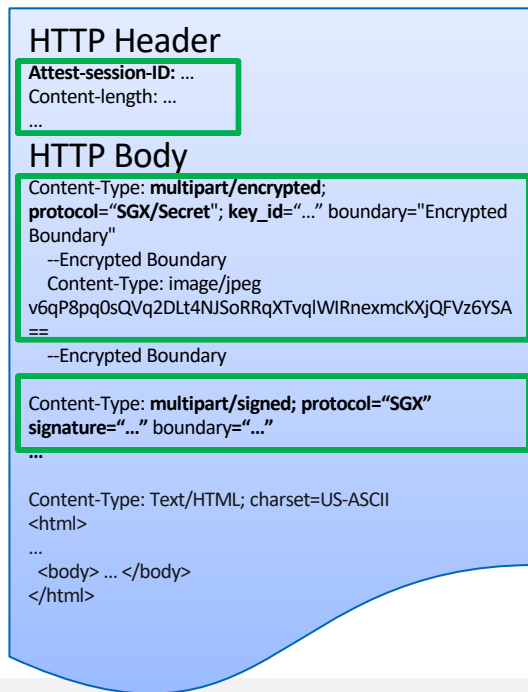
HTTP-Attestable (HTTPA) is an L7 handshake protocol using HTTP extension.

1. Attest-session-ID

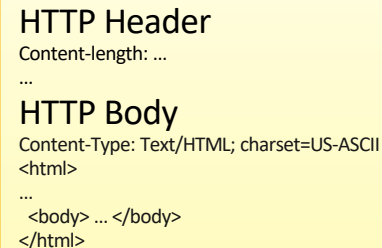
2. Encryption by shared ephemeral keys for TEE sensitive data

3. Signing messages by the ephemeral key

HTTPA

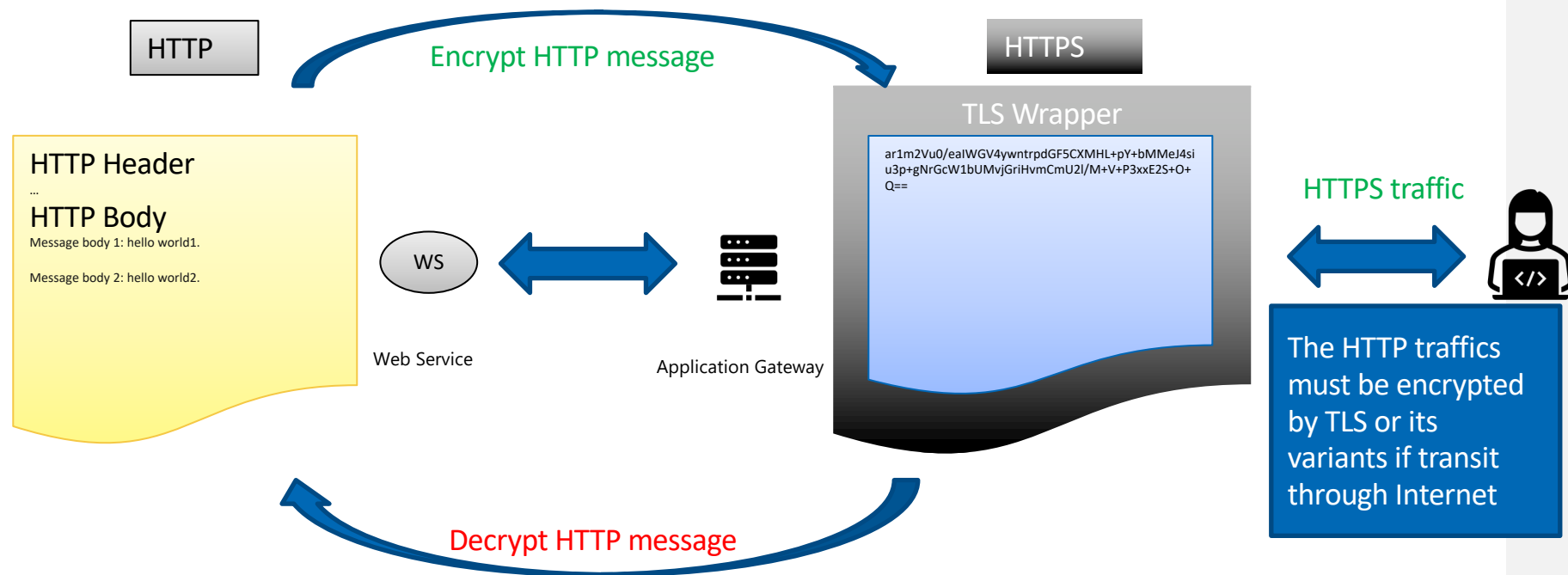


HTTP



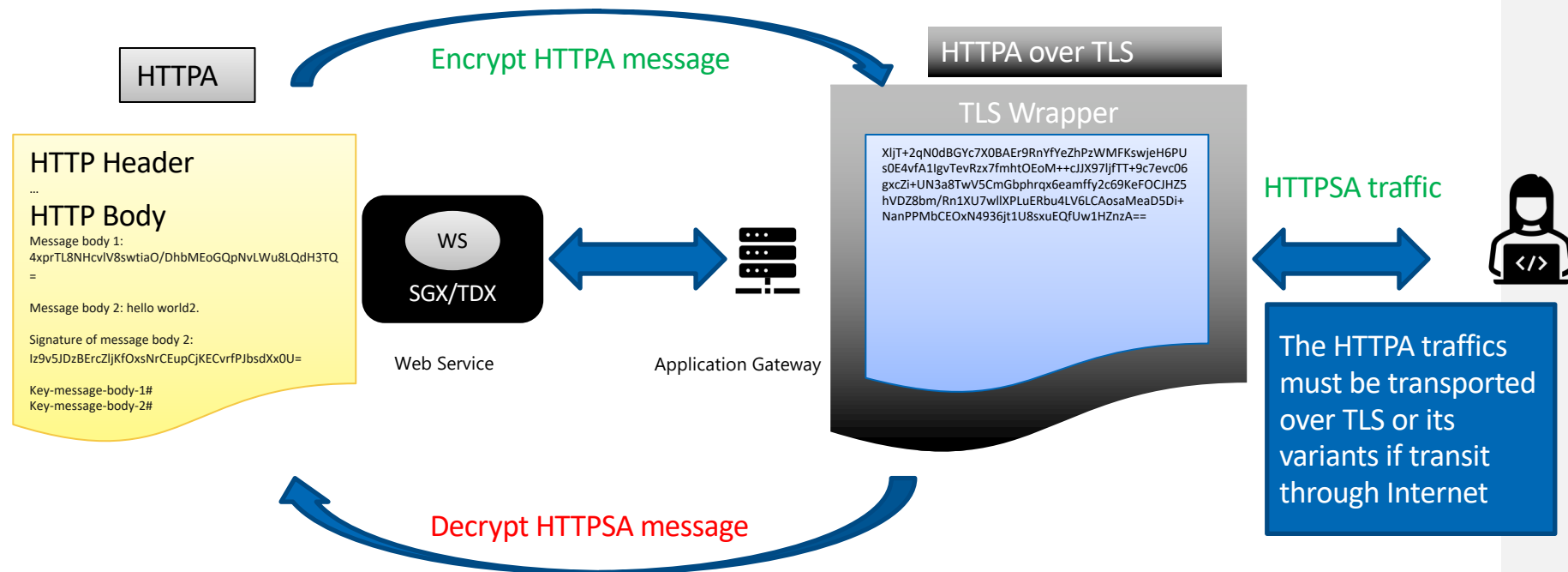
HTTP over TLS (HTTPS)

“Application Gateway supports TLS termination at the gateway, after which traffic typically flows unencrypted to the backend servers” – Microsoft Azure



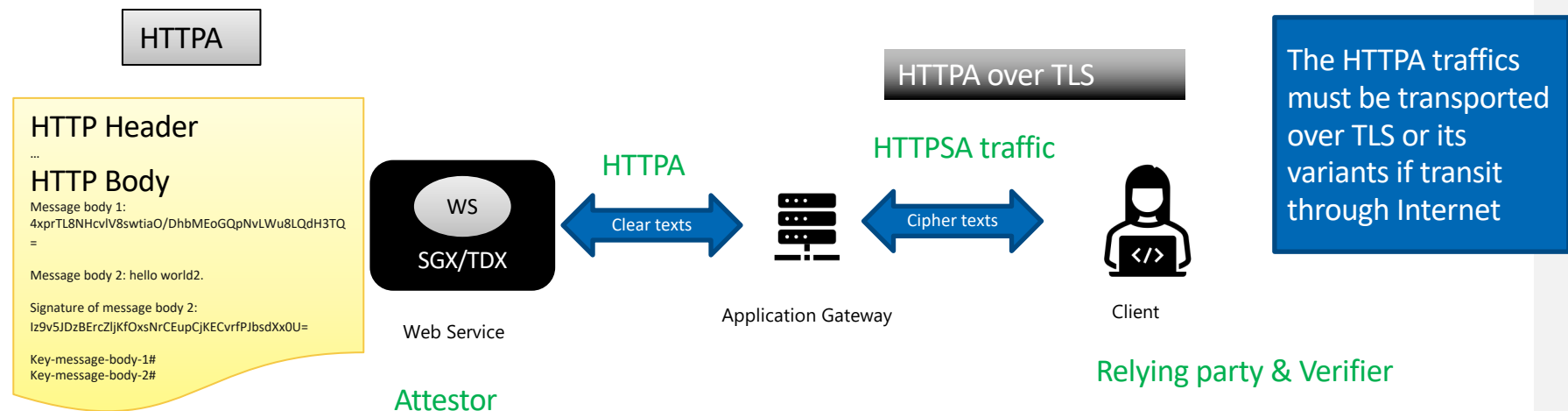
HTTP over TLS (HTTPS)

TEE-encrypted data can go through L7 load balancers without leaking sensitive information!



HTTPSA

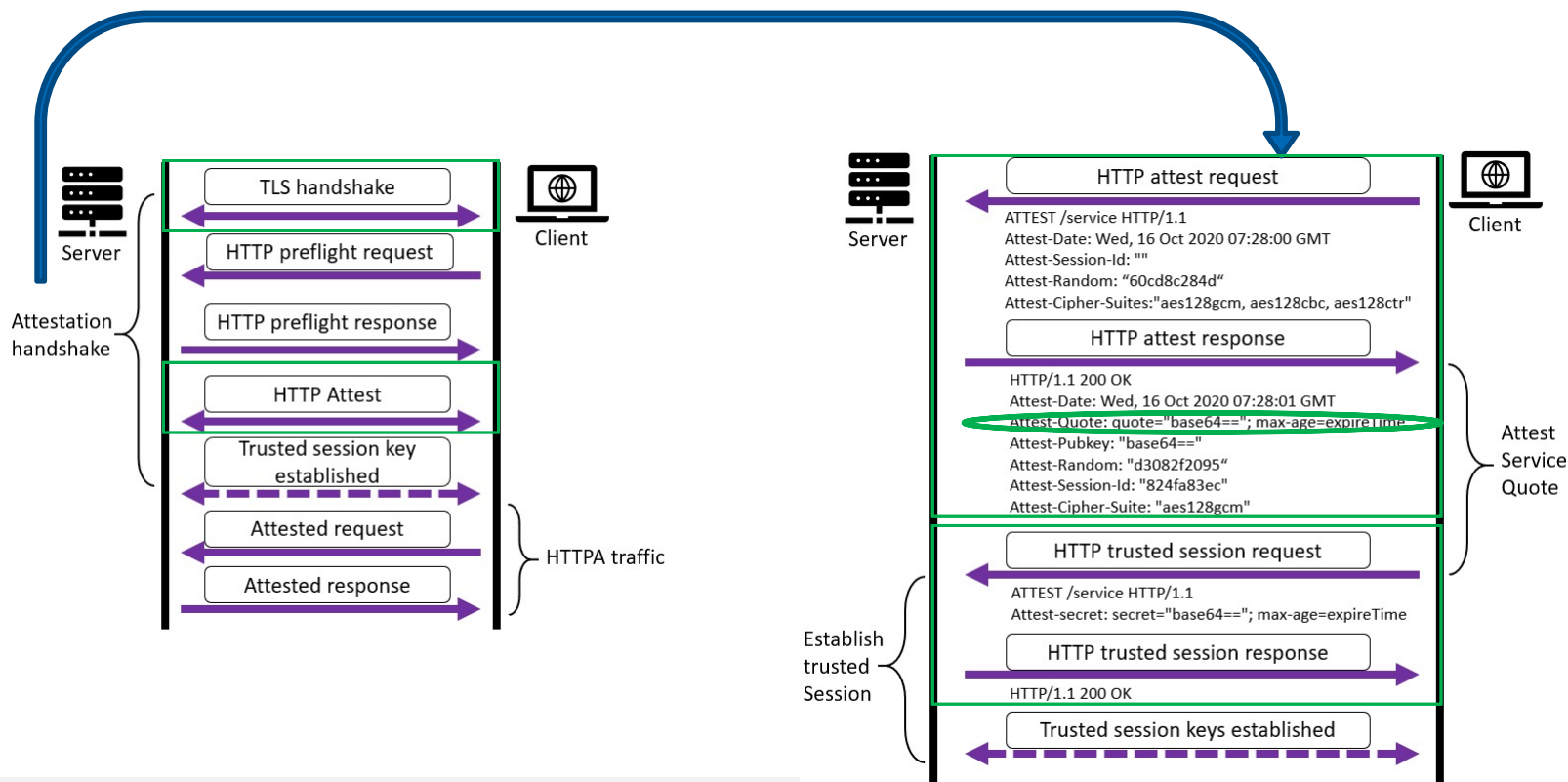
TEE-encrypted data can go through L7 load balancers without leaking sensitive information!



HTTPSA handshake

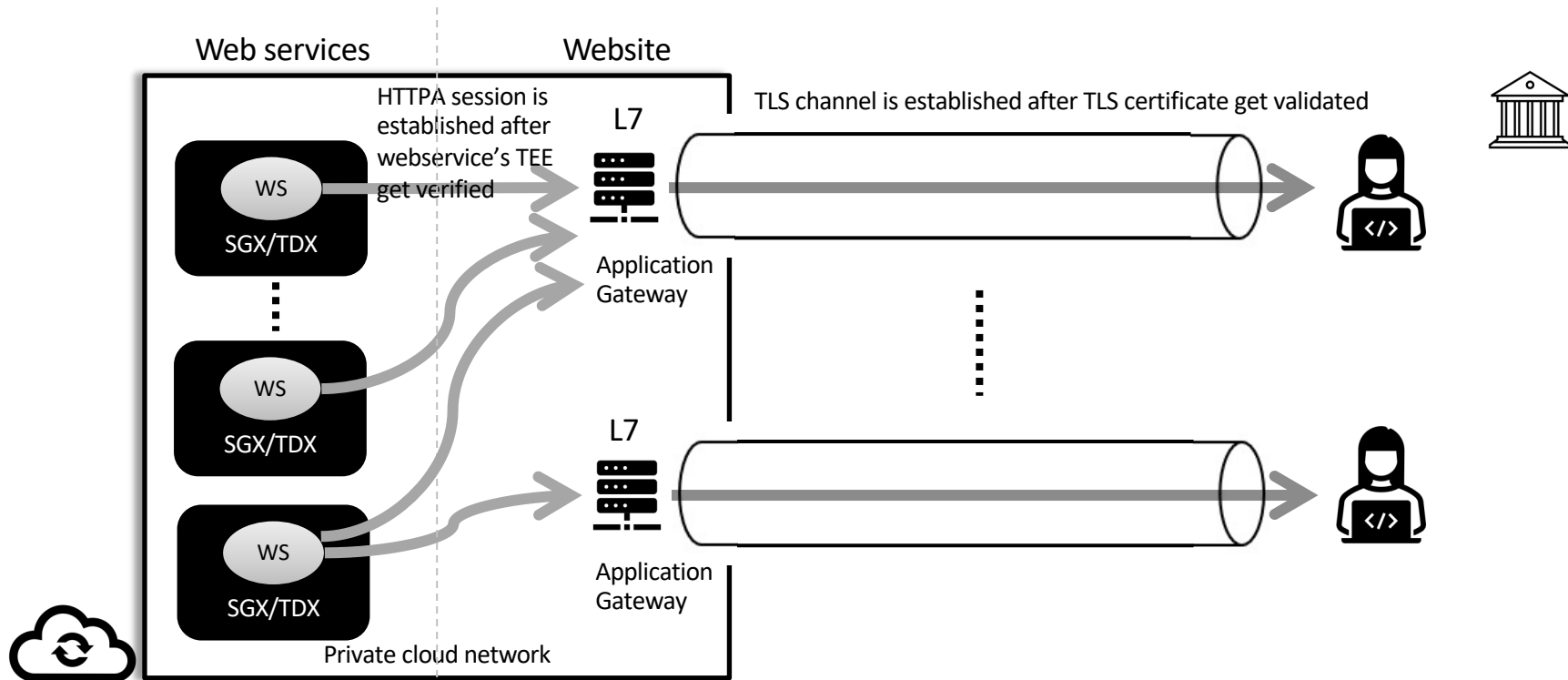
Goal:

1. HTTPSA attestation process
2. HTTPSA key exchange
3. HTTPSA trusted session establishment
4. HTTPSA secret provisioning



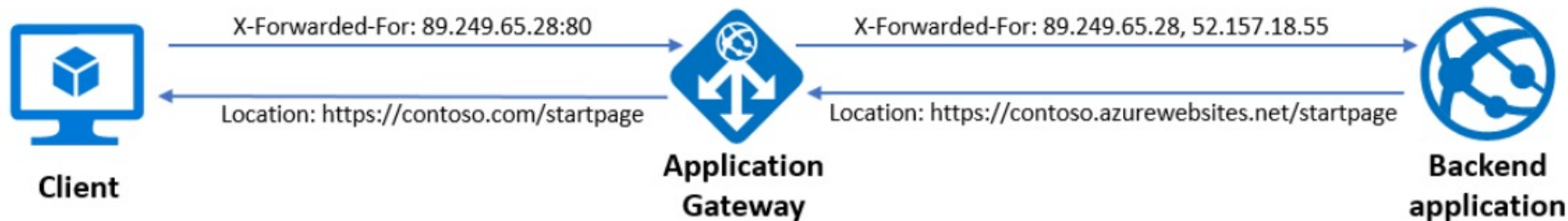
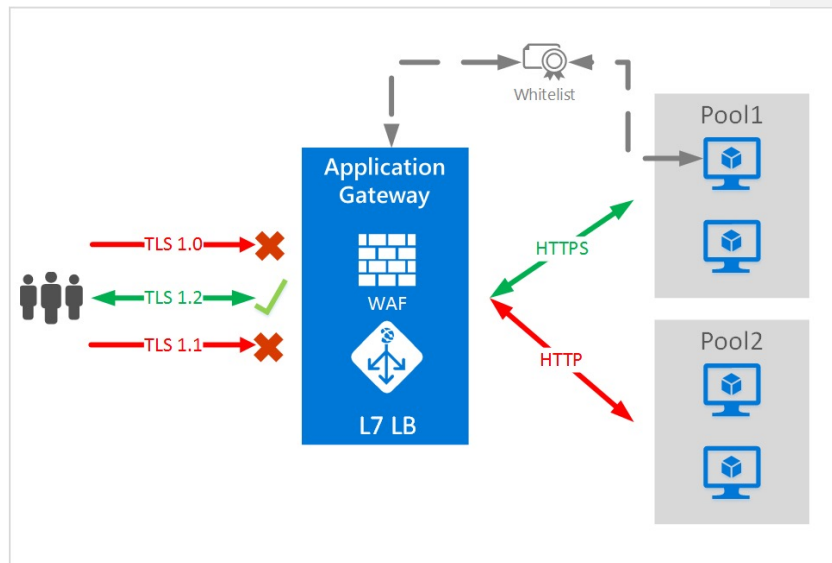
HTTPSA

Trusted end-to-end channel

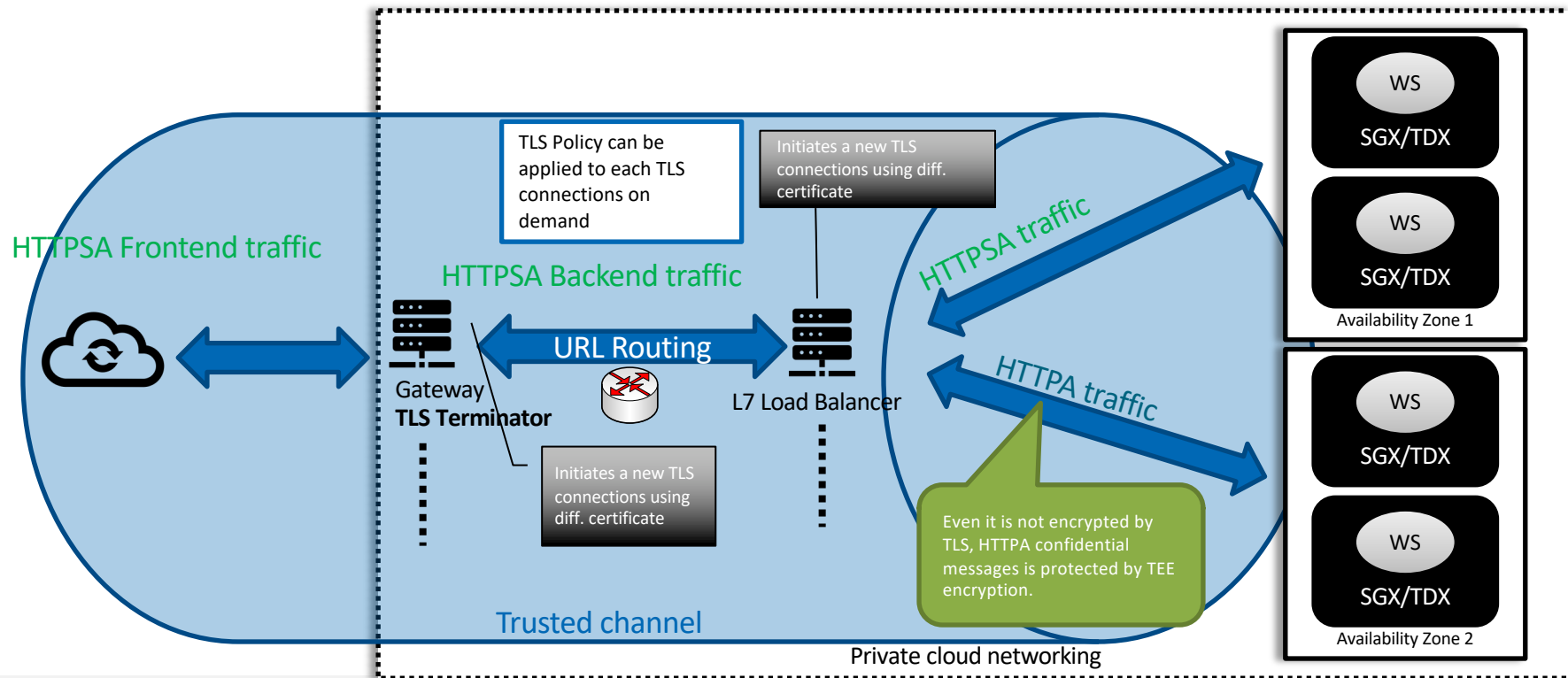


TLS termination is popular

- Better performance.
- Better utilization of backend servers.
- Better routing.
- Better certificate management.

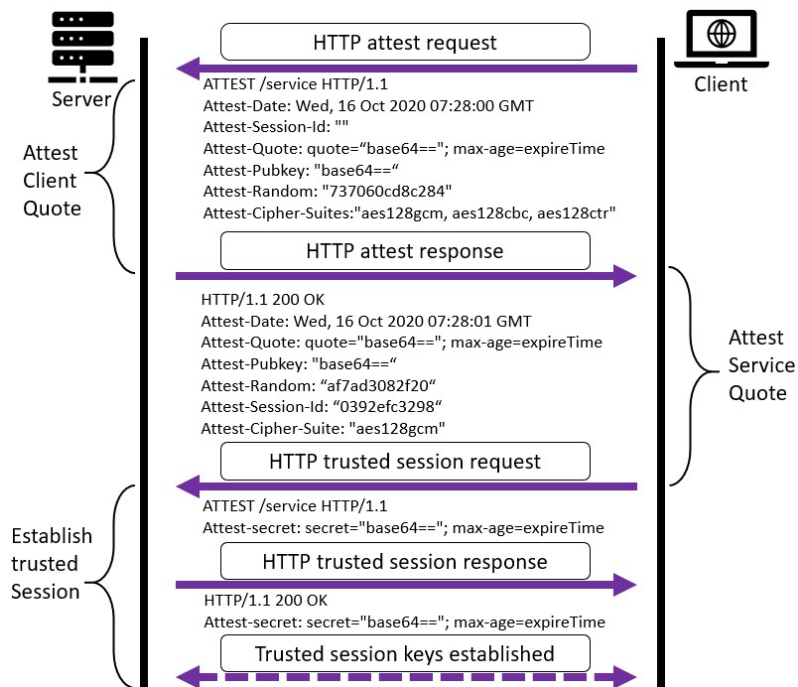


Application gateway and LB with HTTPPA



Mutual-HTTPPA (mHTTPPA)

HTTPPA supports mutual attestation remotely



How does a webservice start using HTTPPA?

- Use HW-TEE to generate HTTPPA quotes
- Implement HTTP extension to support HTTPPA
- Implement web client plug-ins to support HTTPPA

HTTTPA benefits

- Allow users to verify identities:
 - Software service
 - Software vendor
 - TEE
- Establish a TEE-based end-to-end trustworthy channel
 - HTTPS connection + remote attestation + secret provisioning
- Establish TEE for remote confidential computing
 - Execution security
- Allow users for more freedom to control the process over their secrets/data and reject remote services if they do not trust
- Unify remote attestation in a simple way

Feedback of HTTPA/1 from customers

- Customers want perfect forward secrecy.
 - We recommend ECDHE for HTTPA/1
- Customers want both high security and good performance.
 - We recommend ECDHE
- Complexity:
 - Reduce current complexity:
 - If you have secure end-to-end communication at L7, why do we need TLS which increases another layer of overhead?
- Scalability:
 - Current, usage model for TEE is not scalable for customers.
- To some extent, HTTPA/2 has addressed most concerns above.

HTTPPA vs. RA-TLS vs. HTTPS (Cont.)

#	Differentiators	HTTPPA	RA-TLS	HTTPS
1	OSI layer	Application layer	Session layer	Session layer and up
2	TEE awareness	Generic TEE (e.g. SGX, TDX, TPM)	SGX enclave	No awareness
3	Remote attestation	Yes	Yes	No
4	Workload type	HTTP web services	TLS workloads	HTTP website
5	Modified HTTP handling codes	Yes	No; Yes(if you use provisioned secrets)	No
6	Encryption in transit	Yes, if enabling TLS; encrypted messages	Yes	Yes
7	Website/Gateway CA authentication	Yes, if enabling TLS	No (self-signed)	Yes
8	Allow TLS gateway	Yes	No (skip TLS gateway if RA-TLS is bound with workload)	Yes
9	Allow TLS inspection appliance	Yes (not for TEE-encrypted HTTP messages)	No (if RA-TLS is bound with workload)	Yes

HTTPA vs. RA-TLS vs. HTTPS

#	Differentiators	HTTPA	RA-TLS	HTTPS
10	TLS Lib. required for workload	No	Yes, if it is bound with workload	No
11	Certificate signing required	Yes, if using TLS; No, if no using TLS	Yes (self-signed)	Yes
12	CA issued certificate	Yes, if enabling regular TLS	No, because of self-signed	Yes
13	A single webservice to use multiple TLS connections simultaneously (multiple connections)	Supported	No	No
14	Multiple webservices to share single TLS connection (multiplexing)	Supported	No, if bound with workload	No
15	TLS configurable before or during negotiation	No Limitation	Bound to each workload TCB separately, needs ISVs to get involved	No Limitation
16	TLS upgradable/replaceable	No Limitation	Subjected to each workload TCB separately, needs ISVs to get involved	No Limitation

Summary

- HTTPA is an L7 protocol which defines an HTTP extension to build a trusted communication channel over the **Internet**.
- HTTPA facilitates attesting web services based on TEE.
- HTTPA ties secure communication to attestation context for L7 applications.
- We envision HTTPA as an industry standard, such as RFC, to enable confidential applications and accelerate transformation towards **trustworthy Internet**.

Thank you

Hans Wang: hans.wang@intel.com

Gordon King: gordon.king@intel.com





Amazon's approach

- Title: HTTP Message Signatures
- Issue:
 - Integrity of HTTP message is **not** guaranteed across multiple TLS connections
 - TLS-terminating gateway
 - TLS inspection appliance
- Key summary:
 - Describe a mechanism for signing/verifying digital signatures over components of an HTTP message.
 - Require web application-level signing key to be **separate** from any TLS certificate.

RA-TLS

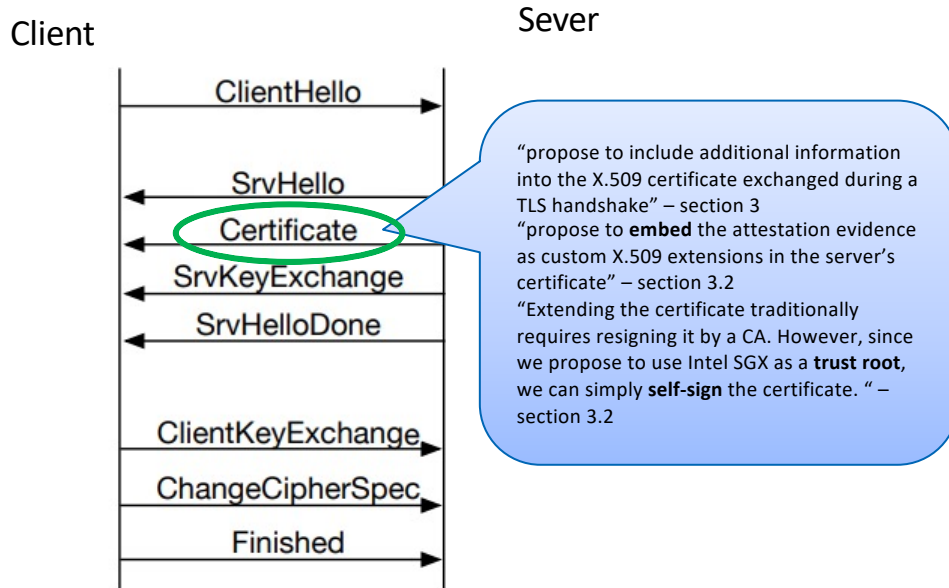
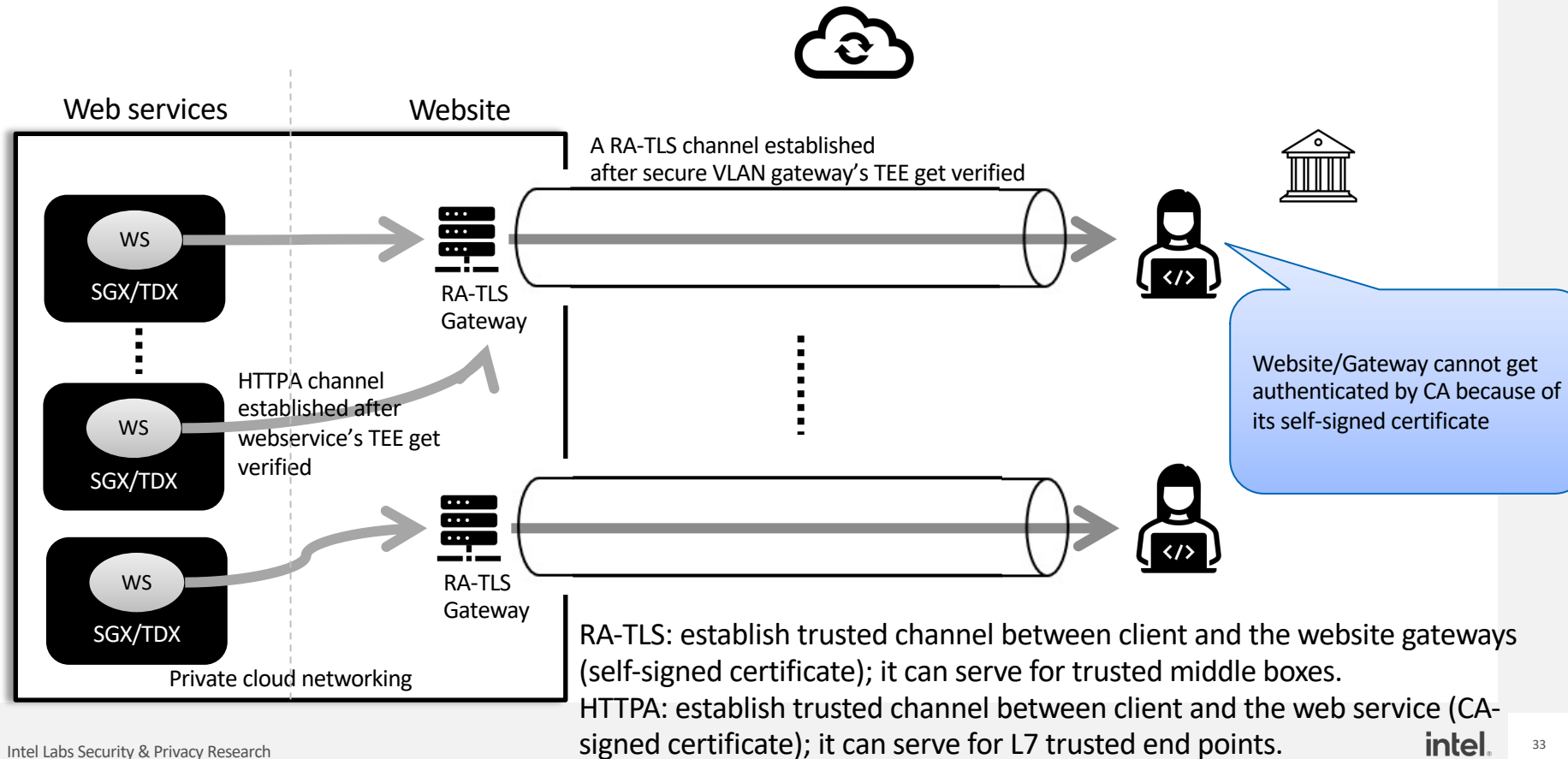


Figure 2: TLS 1.2 Handshake Messages.

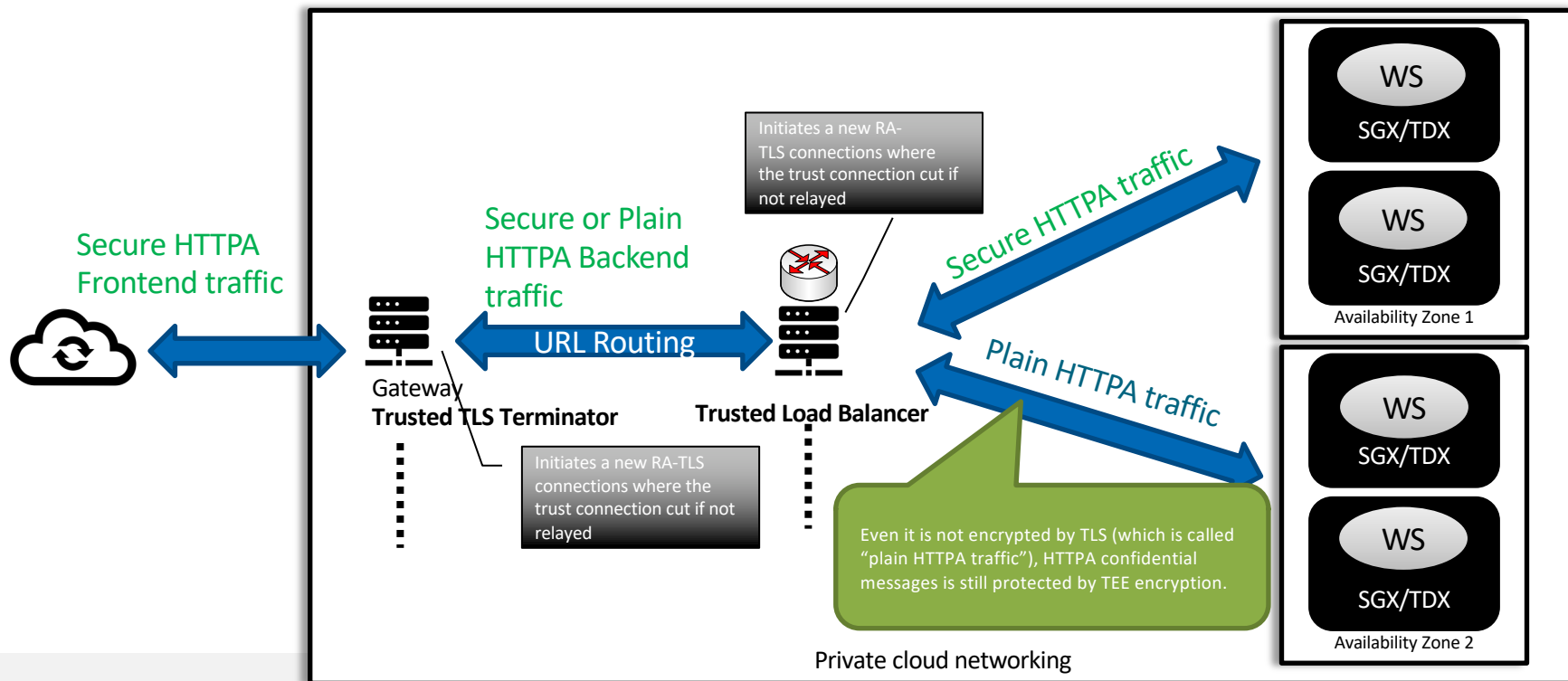
<https://arxiv.org/ftp/arxiv/papers/1801/1801.05863.pdf>

How can we combine HTTPPA and RA-TLS?



HTTPPA Using RA-TLS for Trusted URL Routing

HTTPPA and RA-TLS can co-exist to serve for trusted communication.



OSI Model

