www.connectedthingslab.com

Version 0.1

VeliSphere

Security Mechanisms

Built on Open Standards and Open Technologies

Connected Things Lab

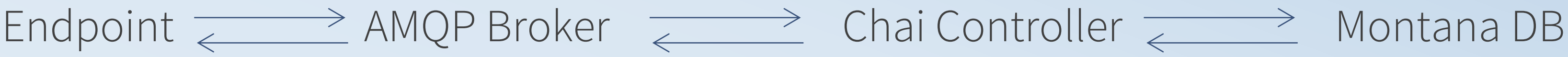
General Message Flow

General Rule:

Endpoints can only send messages to the Chai Controller. Rights management is enforced in RabbitMQ.

Backend systems (Chai Controller, Tigerspice Web Manager, Toucan Web Services) can send to and receive data from any endpoint. Tigerspice and Toucan have their own access rights management to inhibit unauthorized data from being sent to endpoints.

Regular Messages:



Endpoint ⇄ AMQP Broker ⇄ Chai Controller ⇄ Montana DB

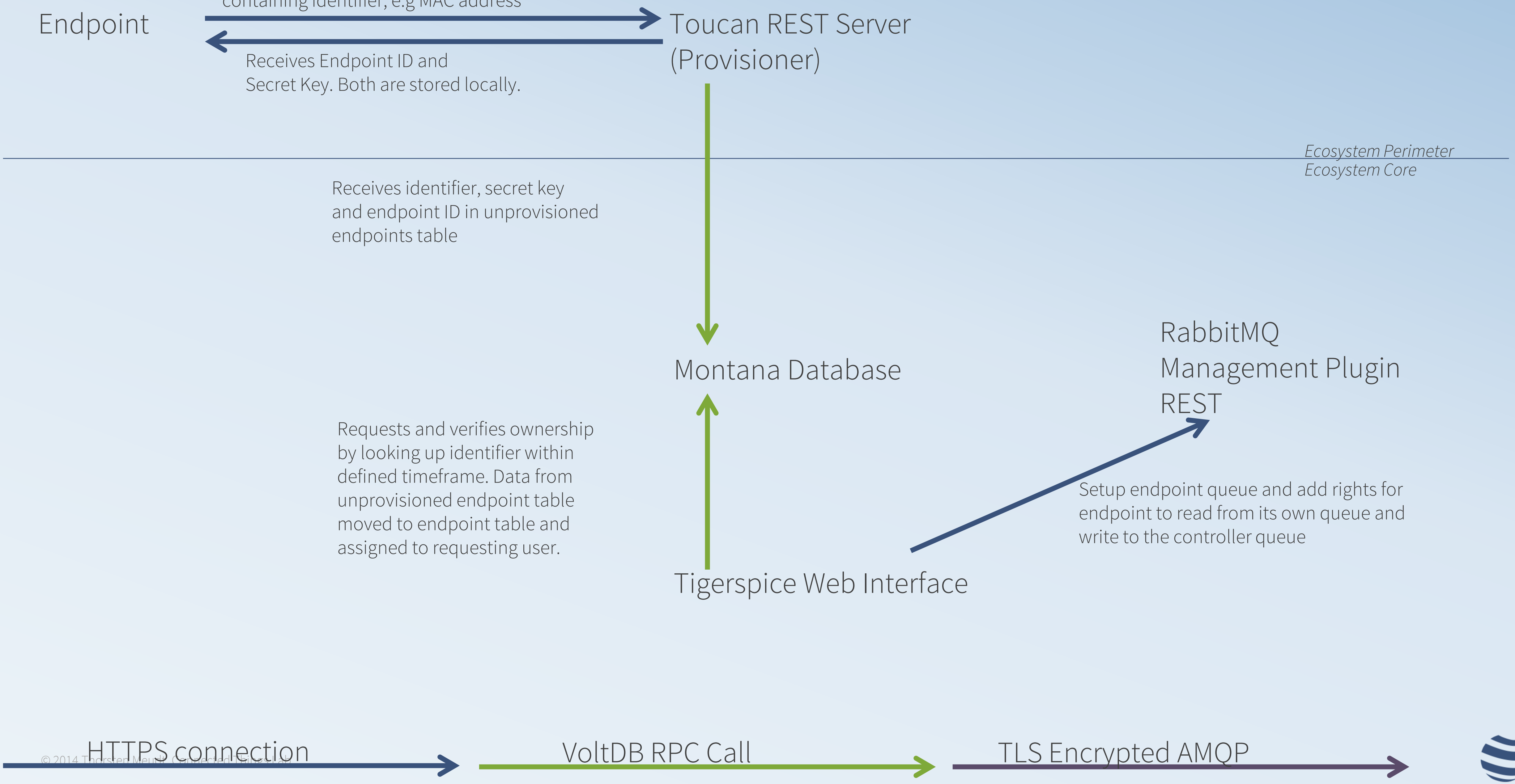
Regular messages contain values from sensors and values sent to actors.
Regular messages can never be sent directly from one endpoint to another.

Endpoint Authentication

HMAC + SSL/TLS

HMAC = keyed-hash message authentication code

PROVISIONING



Endpoint

Sends provisioning request
containing identifier, e.g MAC address

Receives Endpoint ID and
Secret Key. Both are stored locally.

Toucan REST Server
(Provisioner)

Ecosystem Perimeter
Ecosystem Core

Receives identifier, secret key
and endpoint ID in unprovisioned
endpoints table

Montana Database

RabbitMQ
Management Plugin
REST

Requests and verifies ownership
by looking up identifier within
defined timeframe. Data from
unprovisioned endpoint table
moved to endpoint table and
assigned to requesting user.

Setup endpoint queue and add rights for
endpoint to read from its own queue and
write to the controller queue

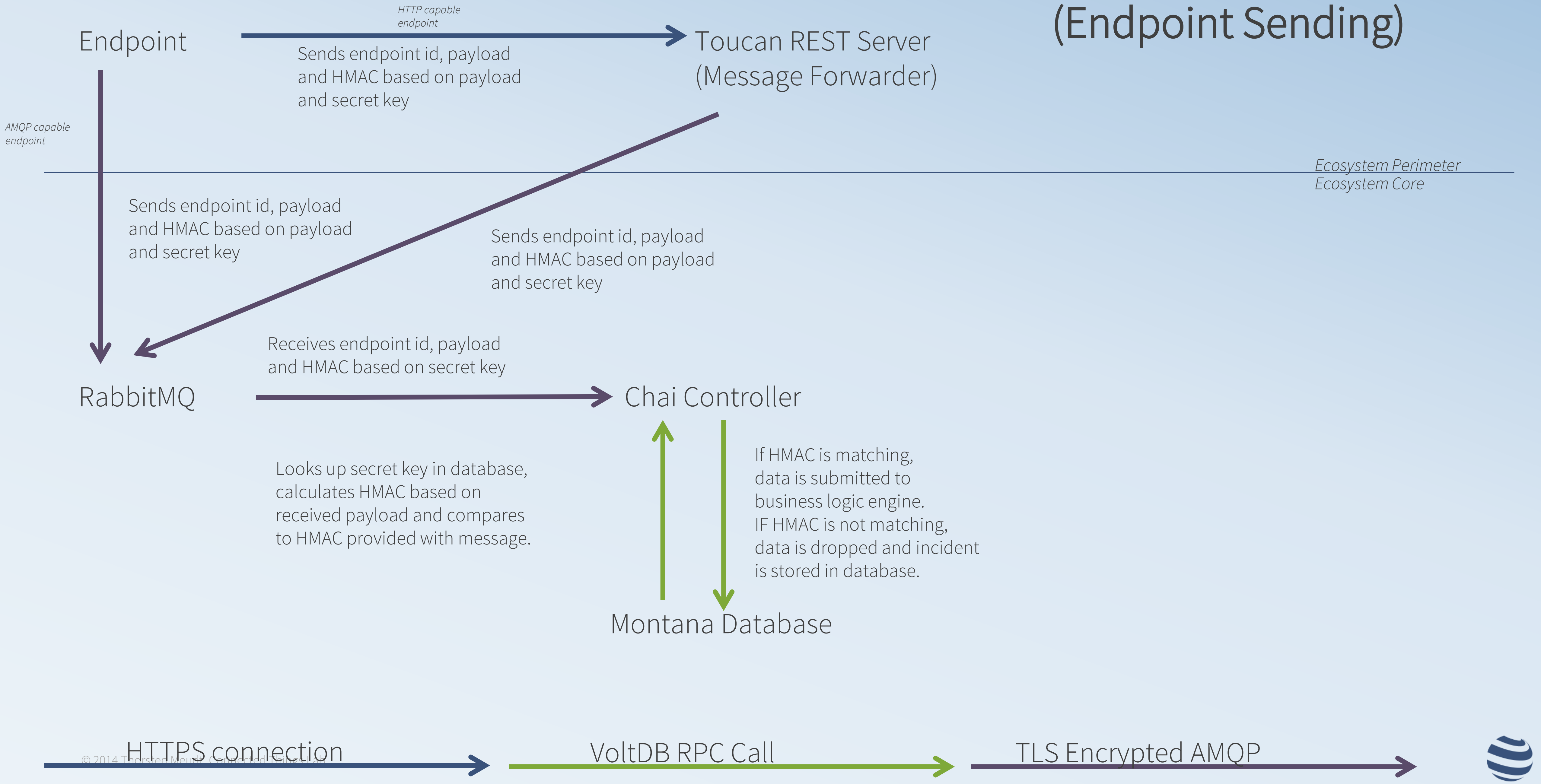
Tigerspice Web Interface

HTTPS connection

VoltDB RPC Call

TLS Encrypted AMQP

AUTHENTICATION
(Endpoint Sending)



Endpoint

HTTP capable endpoint

Sends endpoint id, payload and HMAC based on payload and secret key

Toucan REST Server (Message Forwarder)

AMQP capable endpoint

Ecosystem Perimeter

Ecosystem Core

Sends endpoint id, payload and HMAC based on payload and secret key

Sends endpoint id, payload and HMAC based on payload and secret key

RabbitMQ

Receives endpoint id, payload and HMAC based on secret key

Chai Controller

Looks up secret key in database, calculates HMAC based on received payload and compares to HMAC provided with message.

If HMAC is matching, data is submitted to business logic engine.
IF HMAC is not matching, data is dropped and incident is stored in database.

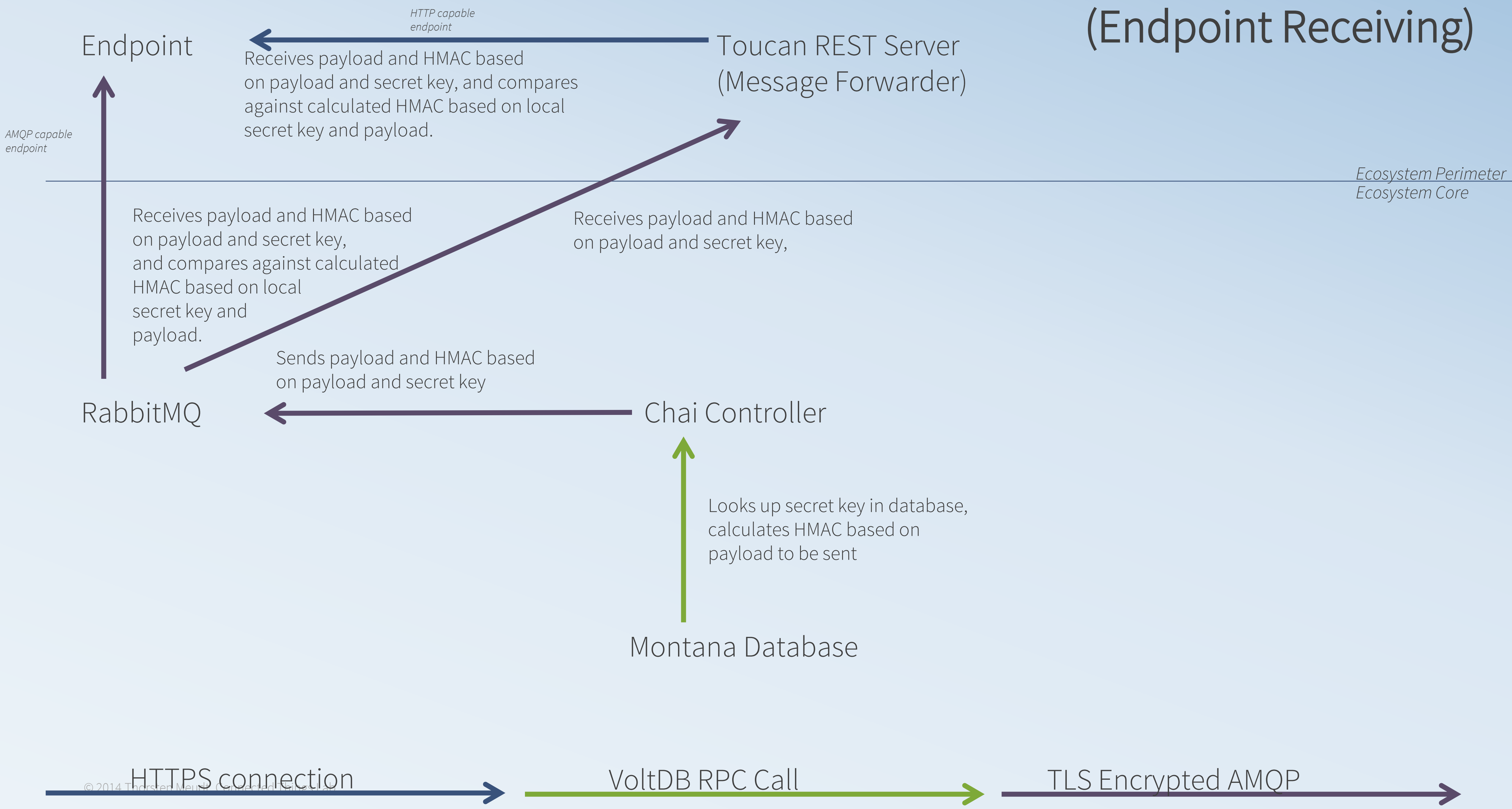
Montana Database

HTTPS connection

VoltDB RPC Call

TLS Encrypted AMQP

AUTHENTICATION
(Endpoint Receiving)



Endpoint

HTTP capable endpoint

Toucan REST Server (Message Forwarder)

Receives payload and HMAC based on payload and secret key, and compares against calculated HMAC based on local secret key and payload.

AMQP capable endpoint

Ecosystem Perimeter

Ecosystem Core

Receives payload and HMAC based on payload and secret key, and compares against calculated HMAC based on local secret key and payload.

Receives payload and HMAC based on payload and secret key,

Sends payload and HMAC based on payload and secret key

RabbitMQ

Chai Controller

Looks up secret key in database, calculates HMAC based on payload to be sent

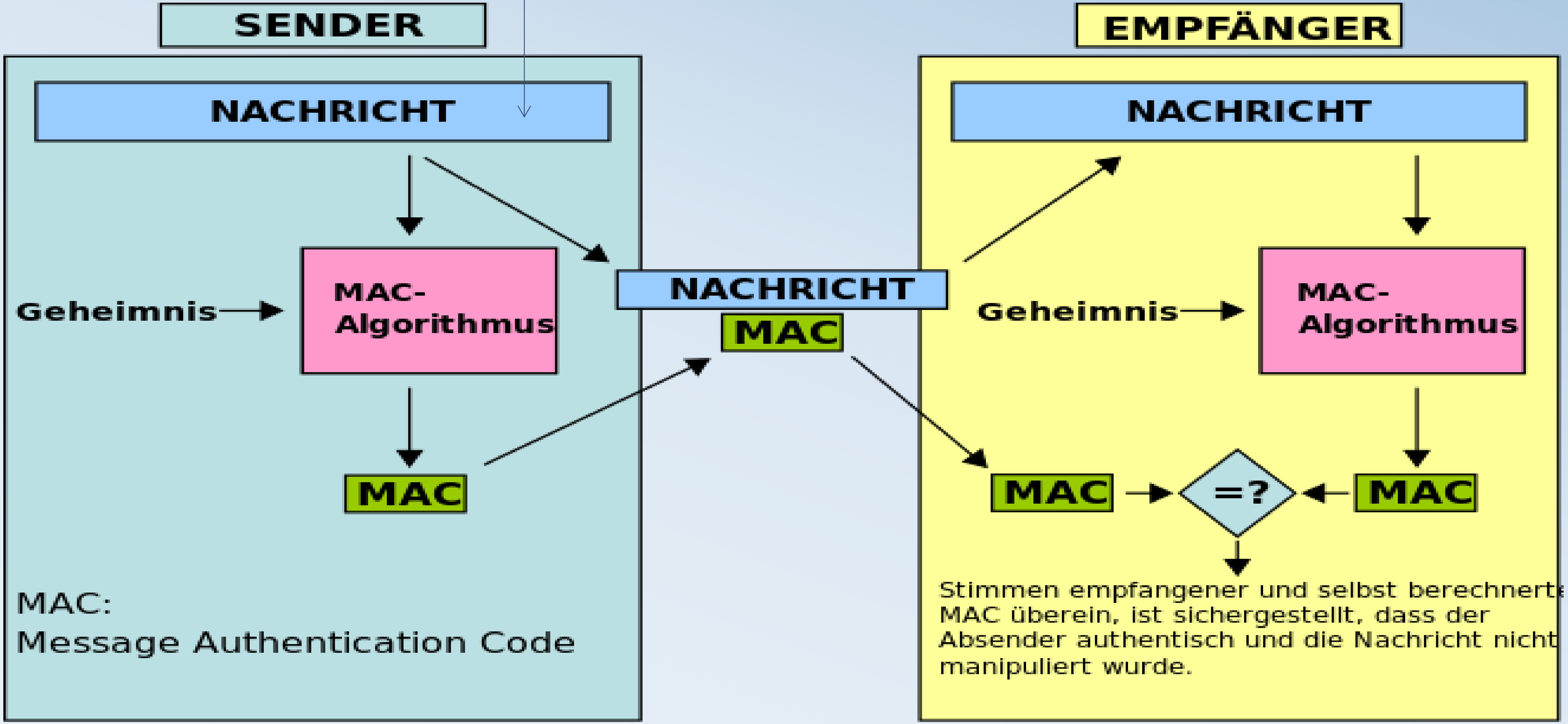
Montana Database

HTTPS connection

VoltDB RPC Call

TLS Encrypted AMQP

Inner JSON



MAC:
Message Authentication Code

Stimmen empfangener und selbst berechnerte MAC überein, ist sichergestellt, dass der Absender authentisch und die Nachricht nicht manipuliert wurde.

API User Authentication

- On initial registration
  - User registers with e-mail and password
  - Random internal UserID and API key are generated and stored in user table
- On API authentication call (via REST)
  - Application calls TOUCAN authentication server, providing user name and password
  - TOUCAN verifies password and verifies if API key is older than 24h → if so, new random API key is generated (to be implemented)
  - TOUCAN responds with USERID and API key
- On API „get“ access
  - With HTTP GET call, Application provides USERID and HMAC calculated based on USERID and API Key
  - TOUCAN verifies if HMAC is matching and responds with payload and HMAC calculated based on payload and API Key
- On API „put“/“post“ access
  - With HTTP PUT call, Application provides USERID, payload and HMAC calculated based on payload and API Key
  - TOUCAN verifies if HMAC is matching and responds with „OK“

Authentication Flow for REST API