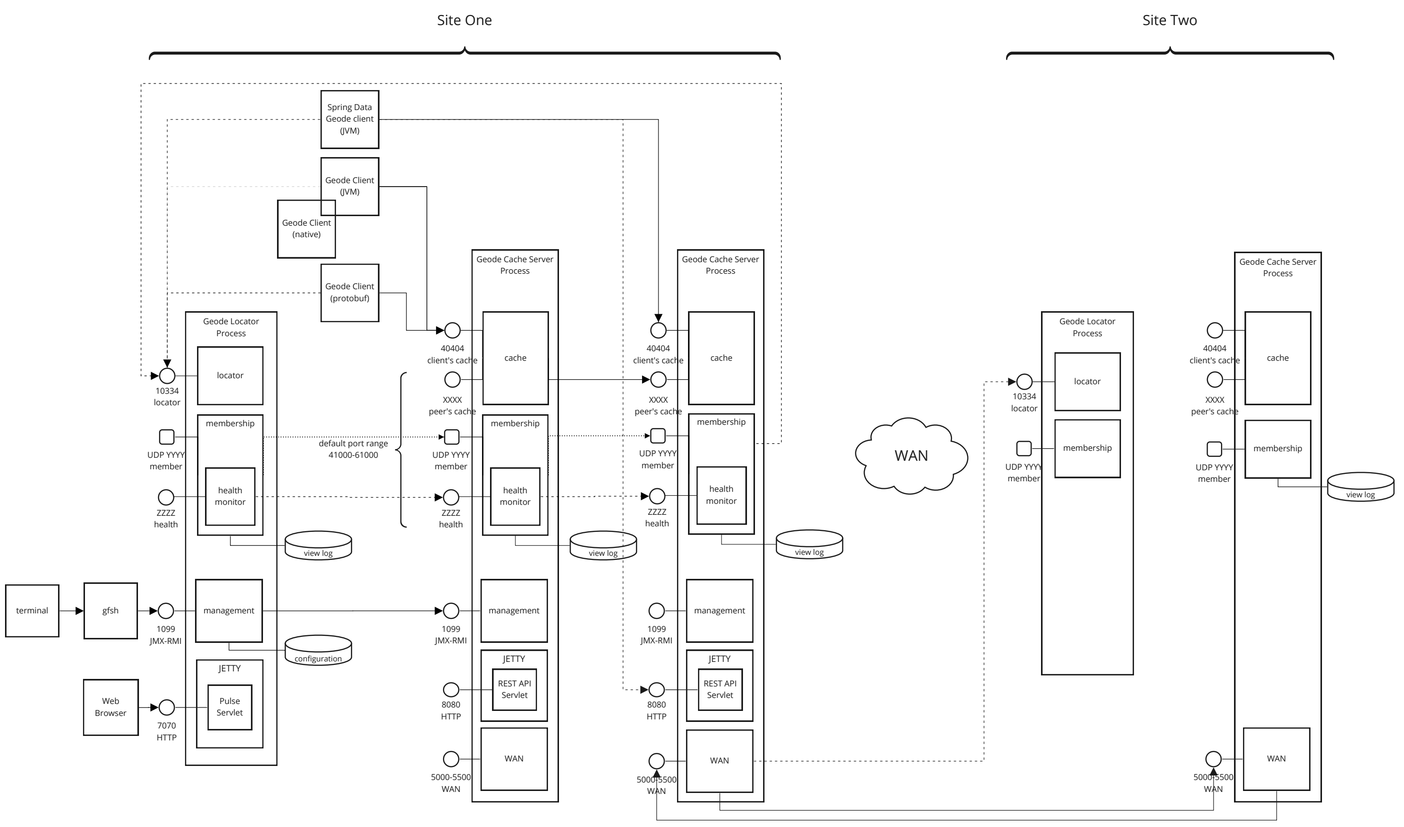

Communication Modes

async

- **UDP unicast** e.g. member interface

request-response TCP full-duplex

- **persistent connection** pooling implemented e.g. client's cache interface, WAN interface
- **ephemeral connection** single request-response per connection e.g. locator interface

hybrid usually async with sync possible too!

- **P2P** e.g. peer's cache interface is usually TCP half-duplex but can optionally use full-duplex

streaming

- CQ and "register interest" results flow from client's cache interface back to client on a dedicated (separate) connection

Serialization Formats

Java Formats

DSFID DataSerializableFixedID
- used for member, and peer's cache interfaces

Java Serializable
- for user-defined cache content

Java, C++, and .Net Formats

DataSerializable
- used in geode-core for peer's cache interface and also for user-defined cache content

PDX
- for user-defined cache content

Language-Agnostic Formats

protobuf
- an alternate format supported by parts of the locator and client's cache interfaces

locator interface (**request-response** with **ephemeral connection**):

"peer" locator interface as defined in **geode-membership** module:

- find coordinator
- get view

+ locator interface defined in **geode-core** module:

- locator list
- client connection
- queue connection
- client replacement
- get all servers
- locator status
- info
- JMX manager locator
- shared configuration status

+ locator interface WAN Edition™ (as defined in **geode-wan** module):

- remote locator join
- locator join
- remote locator ping
- remote locator

member interface (**async** via **UDP unicast**):

- final check passed
- heartbeat
- heartbeat request
- install view
- join request
- join response
- leave request
- network partition
- remove member
- suspect members
- suspect request
- view ACK

health a.k.a. *failure detection* interface (**request-response** with **ephemeral connection**):
- caller requests with (serialization version, and expected view id, and UUID of recipient)
- receiver responds with OK = 0x7B or ERROR = 0x00

client's cache interface (**request-response** with **persistent connection**)

- PutOp
- GetOp
- 73 other Op implementations defined in oag.cache.client.internal and their corresponding Command implementations
- **streaming** CQ and "register interest" results flow to client on dedicated (separate) connection

peer's cache interface (**hybrid P2P**):

- PutMessage
- GetMessage
- 361 other Message implementations

WAN interface (**request-response** with **persistent connection**):

- GatewaySenderBatchOp / GatewayReceiverCommand
- PingOp (from client's cache operators)

Interfaces

Interface and Serialization Format Selection

- locator and client's cache interfaces support Geode message object or protobuf format based on magic number in request

Not Pictured

1. memcached-compatible interface (default port 11211)
2. redis-compatible interface (default port 6379)

author: Bill Burcham
bill.burcham@gmail.com
bburcham@vmware.com
last revision: 8/7/2020