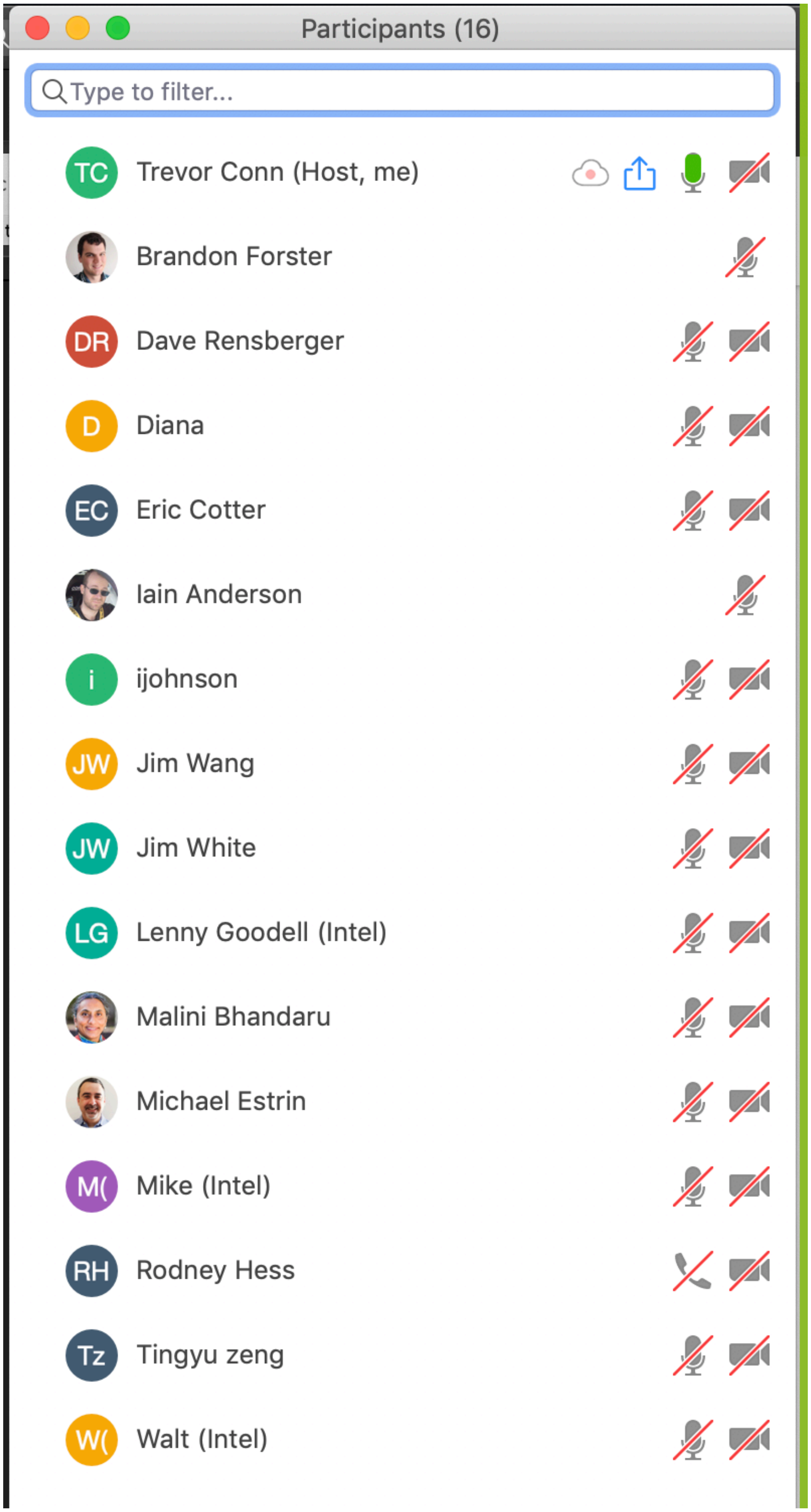
# Core Working Group Meeting Notes (8-Aug-2019)

Attendees:



Trevor Conn (Host, me)

Brandon Forster

Dave Rensberger

Diana

Eric Cotter

Iain Anderson

ijohnson

Jim Wang

Jim White

Lenny Goodell (Intel)

Malini Bhandaru

Michael Estrin

Mike (Intel)

Rodney Hess

Tingyu zeng

Walt (Intel)

**Old Business**

- Consul API import
  - Lenny verified functionality using consul/api module
  - PR merged
    - https://github.com/edgexfoundry/go-mod-registry/pull/25

**New Business**

- Integrating Security Services w/Core Services (40 min)
  - High-level review of integration plan
  - Diana’s questions for the group
    - *Go-mod-core-security connecting to Vault*
      - *Should it use a token? If so, where is it?*
      - *Tingyu – yes, get the location of the master token file from configuration*
      - *Roadmap down the road will be to have a different token for each microservice*
      - *Design stage for secure token distribution service (not current release) – Bryon, Jim Wang*
        - *Hopefully finalized when Fuji is released.*
    - *Init-mongo needs a token to talk to Vault. All services use the same certificate. The token identifies each service individually.*
      - *Application tokens and certificates are stored on the file system*
      - *Definitions*
        - *Token is a service-identifying credential*
          - *Right now this is common to all microservices*
        - *Certificate establishes secure TLS layer*
    - *Is there overlap between “vault-worker” and “go-mod-core-security”?*
      - *“Vault-worker” to be renamed as “security-secret-store-setup”*
        - *Used as one-time initialization for Vault*
        - *Different purpose from go-mod-core-security*
        - *Creates database credentials and stores them into Vault*

      - *"go-mod-core-security" used for Vault integration by other services*
    - *Role of init-mongo*
      - *If security enabled (default) read the DB credentials from Vault and set up Mongo*
      - *Question RE: securing of default Mongo credentials (as in the creds used during the actual installation)*
        - *Will need to investigate….*
    - *Vault API*
      - *We want to use a hashicorp Vault module in the same manner as we've done with Consul for the registry*
      - *Be mindful of dependencies. If there are too many, we won't use it. We'll have to use stdlib Http calls.*
  - Drill into details, update plan based on answers to above
- Discussion on existing PRs (15 min)
  - #1578 – PKI Init
    - Is this ready for merge?
    - This comment makes it sound like some new approach will be taken
      - https://github.com/edgexfoundry/edgex-go/pull/1578#issuecomment-519299929
    - *Design is changing so we will close this and new PRs will be created.*
  - #1627 – API Gateway
    - Question RE: configuration backward compatibility requirement
    - Points for not a problem
      - https://github.com/edgexfoundry/edgex-go/pull/1627#issuecomment-519179444
    - Points for being a problem
      - https://github.com/edgexfoundry/edgex-go/pull/1627#issuecomment-519273484

**Integration Plan**

1.) Complete implementation of go-mod-core-security client
  - a. https://github.com/edgexfoundry-holding/go-mod-core-security
  - b. Responsible for get/set secrets in the "secret store" (i.e. Vault currently)
  - c. This will need to be moved out of holding
  - d. Will be used by mongo-init-go, core services, app functions sdk

2.) Mongo-init-go
  - a. https://github.com/edgexfoundry-holding/edgex-mongo-init/pull/1
  - b. Replaces current init_mongo.js script
    - i. Used in docker-edgex-mongo image
    - *ii. Open question on whether to move it into edgex-go*
  - c. Should be extended to support initialization of Mongo using Vault secrets
    - i. If security is enabled (default)
  - d. Propose making this a part of docker-edgex-mongo since that's where init_mongo.js lives currently
    - i. https://github.com/edgexfoundry/docker-edgex-mongo
    - ii. Should we discuss moving this into the mono repo instead?

3.) Each Core Service
  - a. If security enabled (default)
    - i. Load token and certificate
      - 1. Confirm this will be read from file system/volume
      - *2. Jim Wang pointed out we have not yet designed for mutual TLS, server side only*
    - ii. Request service's DB credentials from Vault (HTTPS)
      - 1. Certificate and token are needed for this
      - 2. Implement this connection in the existing Retry() loop
    - iii. Initialize DB client using credentials obtained from Vault

4.) Bootstrap timing
  - a. Core services should be OK due to retry loop w/configurable timeout
  - b. Config-seed integration
    - i. Which of the new security services should have their configurations populated in Consul? All?
    - ii. *TBD Need security group to define which services config-seed should be aware of once bootstrapping methodology is defined.*