IETF 115, London
OAuth WG
November 2022

Atul Tulshibagwale
CTO, SGNL
Twitter: @zirotrust
LinkedIn: @tulshi

FTA: SGNL Proposal

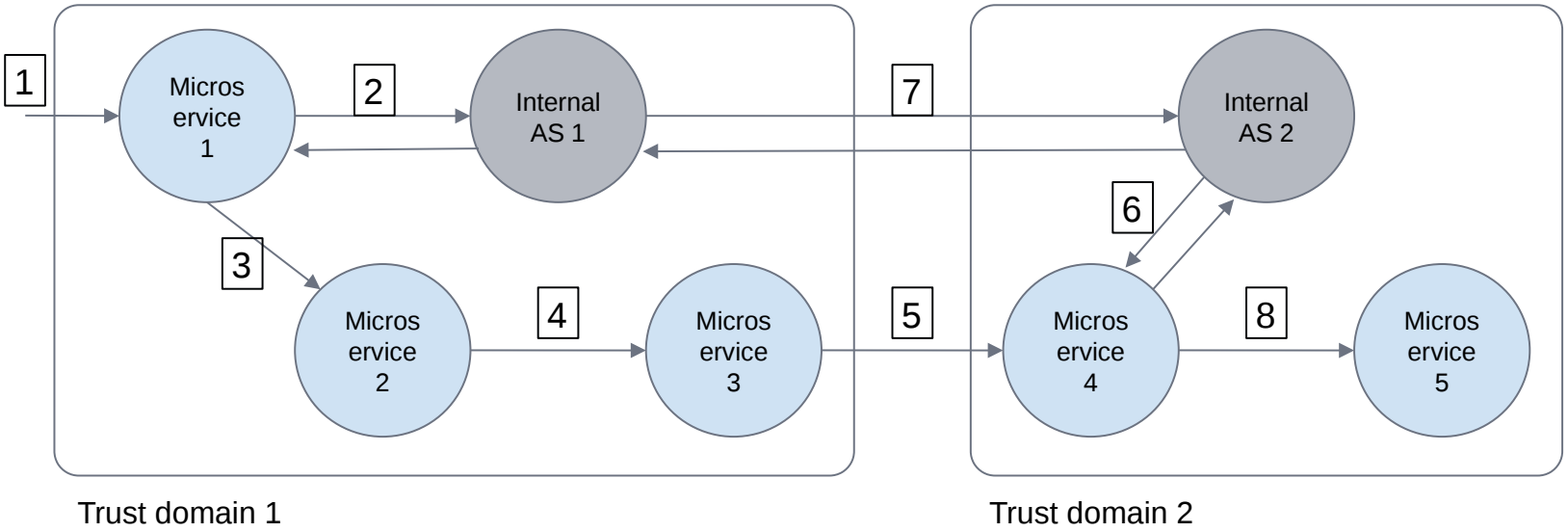
Image credit: worldatlas.com

sgnl

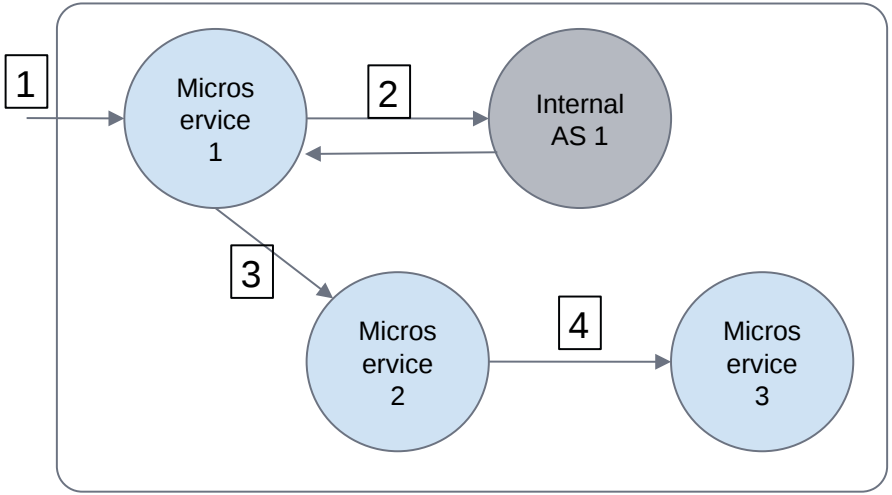
FTA Motivation

- Securing authorization and identity information in microservice communication
- Defense against software supply chain and privileged user compromise attacks
- Needs open-standard to work across multiple cloud platforms and multiple on-premise technologies

Solution Architecture



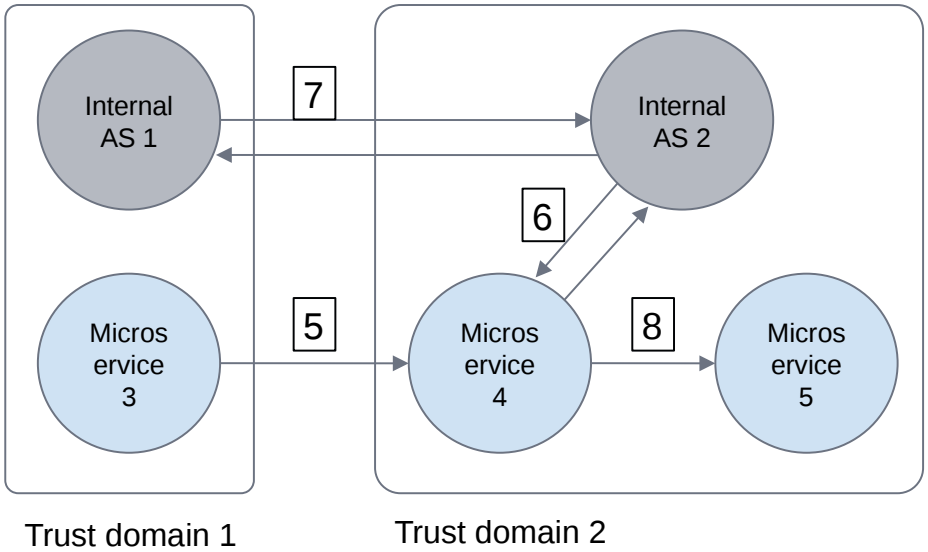
Solution Flow 1 of 2



Trust domain 1

1. MS1 receives external API call with OAuth / OIDC token
2. MS1 obtains IAS1 signed FTA token with context, scope and identity
3. MS1 signs FTAT and invokes MS2 with it. MS2 obtains MS1 public key / certificate from IAS1 to verify token
4. MS2 signs FTAT and invokes MS3 with it

Solution Flow 2 of 2



5. MS3 signs FTAT and invokes MS4

6. MS4 requests FTAT from IAS2, passing the inbound token (note symmetry with step #2)

7. IAS2 verifies TD1 FTAT with IAS1. IAS1 responds with call chain information. IAS2 mints new FTAT with context, identity, AZ scopes and call chain information

8. MS4 signs FTAT and invokes MS5

Note that a service may call out to multiple other trust domains. The flow will remain similar for each such outbound access.

FTA Token Format (Protobuf)

```
import "google/protobuf/any.proto";

message CallChain {
  string domain_id = 1;
  repeated int64 service_ids = 2;
}

message FtaToken {
  int not_after = 1;
  Subject subject_id = 2;
  repeated Scope az_scopes = 3;
  repeated Any context = 4;
  repeated CallChain call_chain = 5;
  repeated Signature fta_sign = 6;
}
```

Notes:

- Subject identifiers and Scope structure need to be discussed
- One call_chain element contains service Ids from only one trust domain. Multiple elements may exist to track the chain across domains
- fta_sign contains signatures of IAS and any microservices that sign the FTA Token
- IAS always has its service_id = 0
- IAS may be selective about which service_ids to include in a call_chain
- Lifetime of a token may be small ~ 5 mins

Signature Format (Protobuf)

```
enum SigAlg {
  SIGALG_UNSPECIFIED = 0;
  SIGALG_RS256 = 1;
  SIGALG_ECDSA = 2;
}

message Signature {
  int64 service_id = 1;
  SigAlg sig_alg = 2;
  string sig_value = 3;
}
```

Notes:

- Trust Domain identifier is always in the call_chain field, not required here

Token Compaction

A microservice may request its internal AS to compact a FTA Token as follows

```
{ not_after = 010101;
  subject_id = “atul@sgnl.ai”;
  az_scopes = [“read”, “write”];
  context = [“ticker_symbol:MSFT”];
  call_chain = {domain_id = “td1”;};
  fta_sign = [{
      service_id = 0; sig_alg = 1;
      sig_value = “very.long.str”;
    },
    {service_id = 1234; sig_alg = 1;
      sig_value = “v.long.str”;},
    {service_id = 5678; sig_alg = 1;
      sig_value = “v.long.str2”;}
  ];
}
```

```
{ not_after = 010101;
  subject_id = “atul@sgnl.ai”;
  az_scopes = [“read”, “write”];
  context = [“ticker_symbol:MSFT”];
  call_chain = {
    domain_id = “td1”;
    service_ids = [1234,5678];
  };
  fta_sign = [{
      service_id = 0;
      sig_alg = 1;
      sig_value = “very.long.str2”;
    }
  ];
}
```

Thank you!