

fabric コマンド直打ち

メトリクス収取

dev-peer のログはデリバリーブロックのタイミングなのか？

エンドースメントプランを取得

```
peer discovery endorsers --channel mychannel --chaincode mycc
```

エンドースメントポリシーの緩和（SDK の promiss が飽和している可能性）

✓ 1. SDK から他の peer に gRPC 接続できるか確認

```
grpcurl -cacert ./orgX/ca.crt -authority
```

```
peer0.orgX.example.com peer0.orgX.example.com:7051
```

```
list
```

100 組織のうち 1 つでも失敗すれば、そこが Proposal 先に含まれた時に submitTransaction() が詰まります

```
OutOf(100, Org1MSP, Org2MSP, ..., Org100MSP))
```

```
try {
```

```
    await contract.submitTransaction('CreateAsset',  
    'asset1');
```

```
} catch (e) {
```

```
    console.error('Error:', e);
```

```
    // 出来ればここで proposal error の内訳を見られるように  
}
```

→ e.responses や e.endorsements に詳細がある場合も

/etc/security/limits.conf で nofile 65536

項目	推奨内容
-----	-----
p-limit で submitTransaction を改善 → 低レベル API で Proposal 先を明示し 並列制御	
Caliper 導入	→ Proposal 失敗原因の可視化・分析に極めて有効

■ Caliper ログで proposal の送信先と応答を確認
caliper launch manager 実行時に --loglevel debug をつける

```
npx caliper launch manager \  
  --caliper-workspace . \  
  --caliper-networkconfig networkConfig.yaml \  
  --caliper-benchconfig benchmarks/simple.yaml \  
  --caliper-bind-sut fabric:2.5 \  
  --caliper-fabric-gateway-enabled true \  
  --loglevel debug
```

2. ログ中に出る proposal 対象と応答のパターンを探す：

[caliper-flow] INFO Caliper has successfully initialized

[transaction-creator] Sending proposal to peers:

peer0.org1, peer0.org2, ...

[transaction-endorser] Received endorsement from
peer0.org1

[transaction-endorser] Received endorsement from
peer0.org2

[transaction-endorser] Endorsement from peer0.org3

failed: context deadline exceeded

→ 各 proposal の送信対象・応答・失敗理由が記録されている

3. 問題のトランザクションで失敗した peer を特定できる

●Grafana

| メトリクス

| 内容

|

| ----- |

| ----- |

|

`fabric_peer_transaction_endorsement_duration_seconds` | proposal→endorsement までのレイテンシ |

| `fabric_peer_transaction_endorsement_failure_total`
| endorsement の失敗回数 |

|

`fabric_peer_transaction_endorsement_success_total`
| 成功回数 |