LTI Dynamic Registration 概要

7/19 15:00~ 日本 1Edtech LTI部会

コニカミノルタ則武

はじめに

デジタル学習環境エコシステムの価値を向上するために、
トータル(関係者間)の運用コストを下げることも重要と考えております。

本日は関係者間の運用コストを下げることを大きなテーマ/目的として、
LTIの仕様の利用が日本で適用可能か？　今後検討を進めるためのきっかけになればと思います。

アウトライン

背景について
　現状、LTI連携時に必要になる情報、課題等

LTI Dynamic Registrationについて
- 概要
- OpenID Connect Dynamic Client Registration について
- LTI Dynamic Registration

今後の検討項目など頭出し
- ユースケース
- 疑問点など

背景

前提

エコシステムとしては目指すのは、Toolが豊富にありさまざまなシステムやアプリケーションを可能な限り自由に選択し、組み合わせて利用できる環境

以下は標準V3.0.0に記載の学校（管理職等）のユースケース

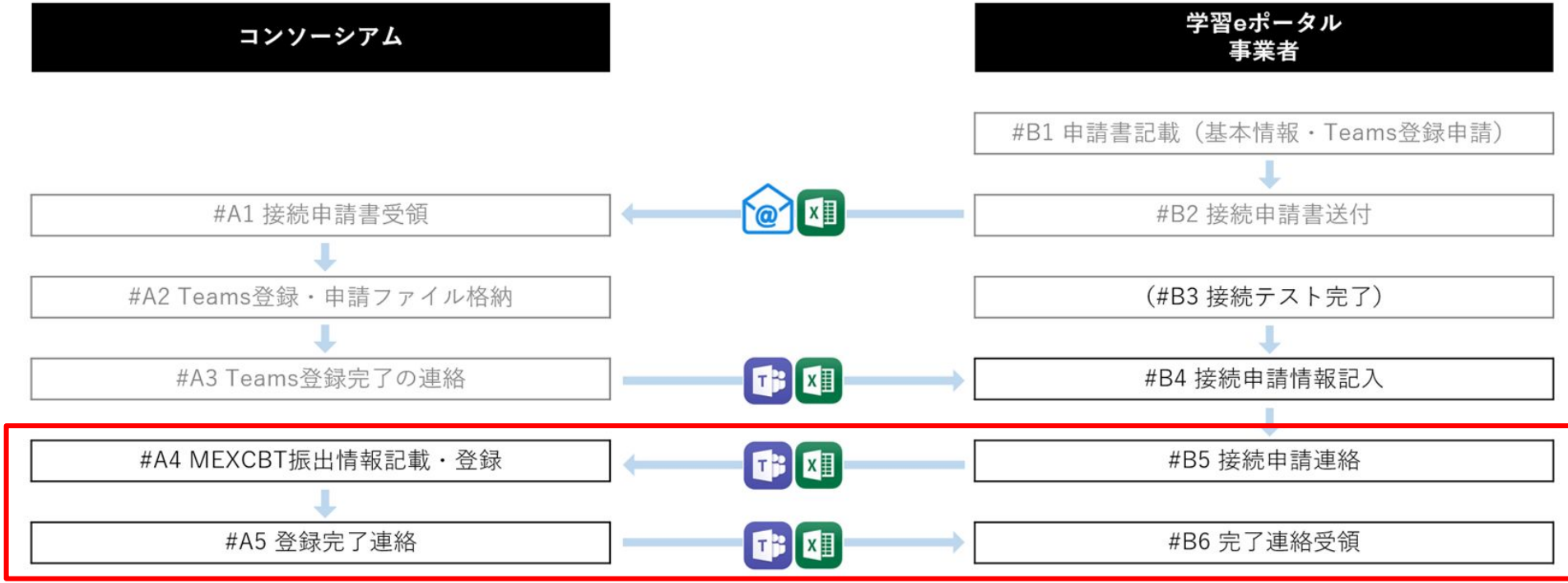
デジタル教材を学習eポータルに登録する	・学校（管理職または教職員等）は、授業や家庭学習における課題等で利用するデジタル教材を学習eポータルに登録し、教科や教科書の単元、時間割等と紐づけを行い、デジタル教科書と連携して利用できるようにする

一方、現状は運用チーム(SRE)での手作業(Toil)が発生している

今後Toolが豊富になり事前設定を行うところにエコシステムとしてトータルコストが発生するのを見越して、事前設定の簡素化を検討する

現状(MEXCBTとの事前設定)

接続申請について



現状(MEXCBTとの事前設定)

LTI連携のための設定スコープに限る（スタディログ、アカウントや学校コード等はスコープ外）
設定値のやり取りと反映に手間がかかる（やり取りと反映にだいたい 1-2週間くらい）

	学習eポータル	MEXCBT(Tool)
①	以下の情報をMEXCBTコンソーシアムへ送付(Teamsにて) ● Issuer ID ● Client ID ● JWKs URL ● OIDC AuthN URL ● OAuth token URL ● [学習eポータルドメイン名] ● [学習eポータル IPアドレス]	学習eポータルからの情報をMEXCBTへ反映(水曜13時までに共有された情報を金曜夕方までに反映)
②	MEXCBTからの情報を学習eポータルに反映(常時、メンテ時等々)	以下の情報を学習eポータル事業者へ送付(Teamsにて) ● JWKs URL ● OIDC Initial URL ● OIDC リダイレクト URL ● DeepLinking Launch URL ● [MEXCBTのIPアドレス]

LTI連携時に事前に必要になる情報 What

LTI連携のための設定スコープに限る（スタディログ、アカウントや学校コード等はスコープ外）
Platform-Originatingの場合(学習eポータルからToolを起動するケース)

	Platform(学習eポータル)	Tool
MUST	● Client ID(Tool) ● OIDC Initial URL	● Issue ID(Platform) ● Platform Public Key(URL or Key自体)
Optional	● OIDC リダイレクト URL ● DeepLinking Launch URL ● Deployment ID	● Client ID ● OAuth token URL(AGS利用時) ● Deployment ID
独自	● 相手の出口IPアドレス ● 相手のドメイン ● IDトークンの最大長 ● その他	● 相手の出口IPアドレス ● 相手のドメイン ● IDトークンの最大長 ● その他

設定のタイミング When

運用中の事前設定1回のみではなく、様々なタイミングで情報のやり取り /反映が発生する

開発時　※開発環境/ステージング環境への設定	頻度
環境追加時(初回や、各種開発/ステージング環境など)	それなりにある
各々のシステムのバージョンアップなどで設定値等変更時	あまりなさそう
バージョンアップで設定値の更新時	あまりなさそう

運用時　※運用環境への設定	頻度
初期構築時	1回はある
Tool追加時	それなりにある
学習eポータル入れ替え時	そこそこある？
バージョンアップで設定値の更新時	あまりなさそう

課題/手間となりえる箇所(予測等含み)

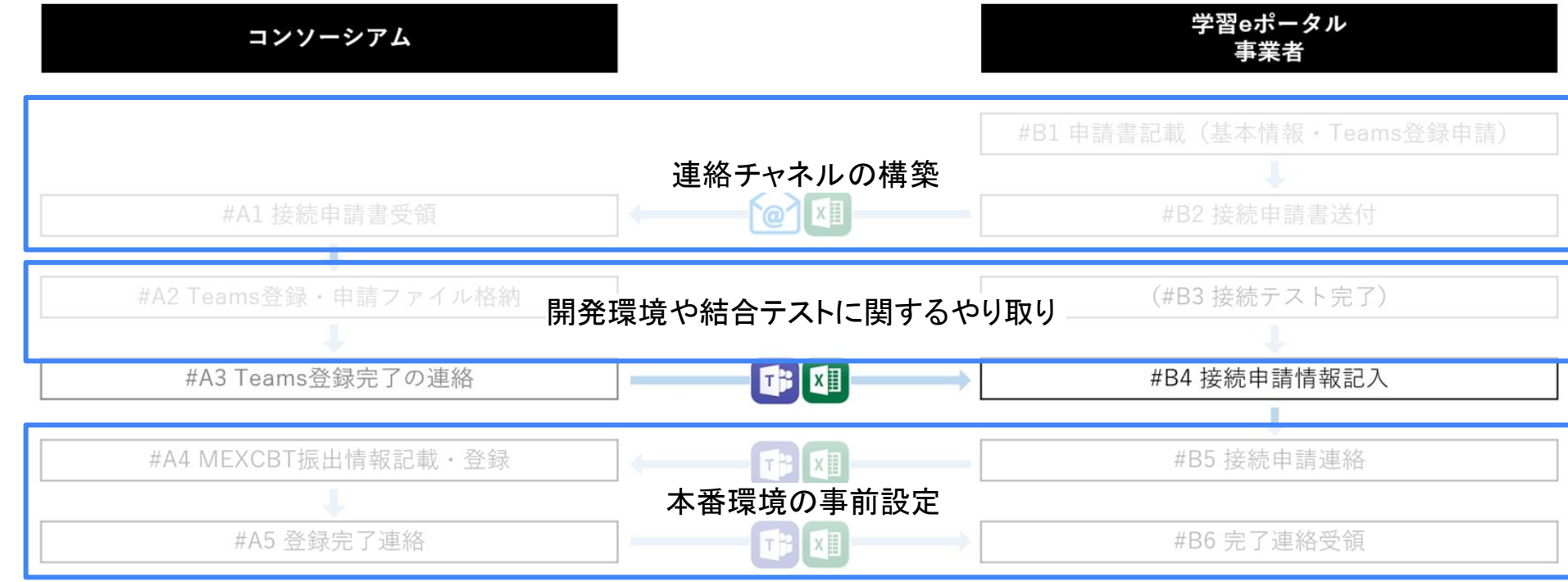
エコシステム過渡期として LTI Dynamic Registrationのみではカバーできなそう
一方、LTI Dynamic Registrationで受けられるメリットはありそう

- LTI連携情報以外の情報のやりとり
  - 通信のための設定値が存在する
  - Firewall系での設定情報などは MEXCBTとの間でも発生している
  - CSPヘッダーの設定 (iframe周りの設定)
  - アプリケーション層以外の設定の動的な反映はできないことはないが、なるべく避けたい
- 必要情報の周知や共有 (カスタムフィールドはなに？)
  - Platform⇔Toolで、Platformごと、Tool毎に異なる情報の把握と払い出し
- 情報のやり取り
  - やり取りするチャネルの準備
  - やり取りする情報のバージョン管理
  - 情報の手違いがあった場合のやり直し (コミュニケーションコスト)
- 設定の反映
  - 設定の人手での反映
  - 設定反映時に他システムに与える影響考慮 /調整コスト

現状(再度みてみて)

現状もLTI連携の設定に至る前に、手間 /コストがある

接続申請について



打ち手案

相談/議論したいポイント

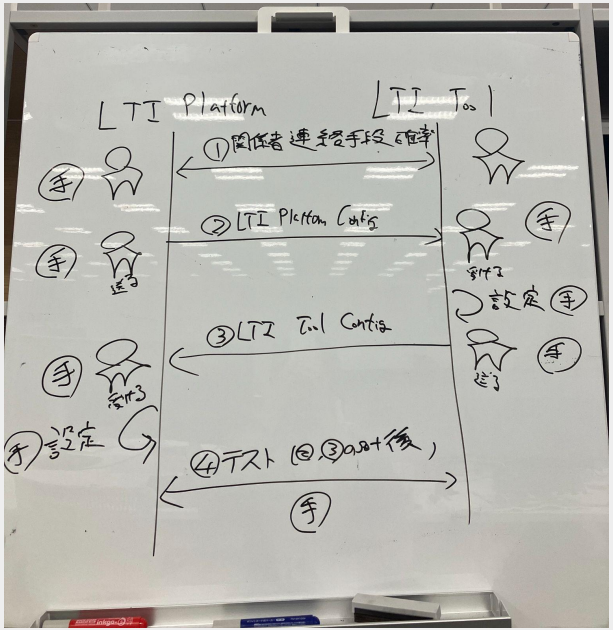
直近の打ち手　エコシステムの成長に合わせて都度対応

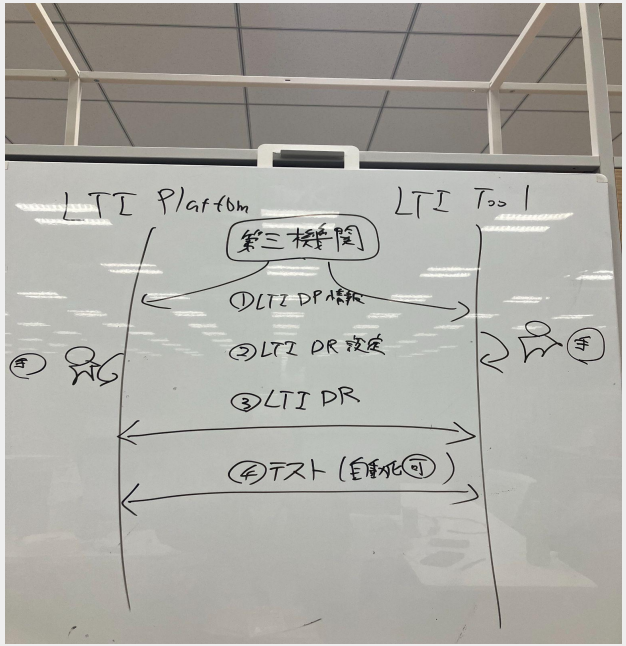
課題	対応	具体
必要情報の周知や共有	登録情報フォーマットを作成 （メタデータなどの記載を含む） パラメータ値やバージョニング、連絡先に等々を含むシート	MEXCBT接続申請書をベースに作成
情報のやり取り	Platform事業者とTool事業者の連絡チャネルの事前準備	TeamsやSlackを用意？ Githubのレポジトリを運用？ (主管は？)
設定の反映	インフラ面などは各事業者ごとの範囲 **LTI(アプリケーション層)の場合は動的に連携も可能**	**LTI Dynamic Registration**

(要検証詳細化)LTI Dynamic Registrationのベネフィット

※時間あれば追記　LTI DR有無での作業フローの比較

手作業でのやり取りや設定が減る　Tool×Platform数の手間





LTI Dynamic Registration概要

技術標準理解：リファレンス

LTI DR Spec. at 1Edtech	http://www.imsglobal.org/spec/lti-dr/v1p0
OpenID Connect Discovery	https://openid.net/specs/openid-connect-discovery-1_0.html
OpenID Connect Dynamic Client Registration	https://openid.net/specs/openid-connect-registration-1_0.html
Dynamic Client Registration Overview	https://curity.io/resources/learn/openid-connect-understanding-dcr/

前置き:スペックの状態について

OpenID Connect Discovery
OpenID Connect Dynamic Client Registration
⇒Final　確定済み

LTI Dynamic Registration
⇒IMS Candidate Final
　まだ変更される可能性あり

LTI DR サマリー

- LTI Dynamic Registrationによって得られるメリットはありそうで、適用することも可能でありそう
- 技術面や運用面などの解像度を上げて、判断や実際の標準化を行っていきたいというステータス
  様々な観点から疑問点懸念等お聞きしまして、今後につなげていく

- OIDCの標準仕様を活用した仕様になっている
- 規定された範囲と規定外の部分がある
  - 規定された範囲内：やり取りする情報　I/F仕様
  - 規定外の範囲：I/F自体の保護の部分　デプロイパターン
- 記述ややり取りされる内容としては十分で、カスタムフィールドもある
- プロトコルの実現可能性については現時点でみれていない

LTI Dynamic Registration概要(先載せ)

- LTI PlatformへのLTI Toolの登録を簡易にするための標準仕様
- OpenID Connect Discovery、OpenID Connect Dynamic Client Registration、[RFC7591] OAuth 2.0 Dynamic Client Registration Protocolの仕様を活用
- OpenID Connect Discovery、OpenID Connect Dynamic Client Registrationと異なる点
  - Platformからトリガーされる点
  - Tool, Platform間で長く関係が続く
  - Platform側でToolを使うか判断する（DCRも同じでは？）
- Metadata Definitions
  - OIDC+LTI連携で必要な情報の記述が可能⇒Platform and Tool Configuration Metadata Definitions
  - カスタムフィールドも持てる
- Registration Protocol
  - Step1 : Registration Initiation Request
    - ToolのRegistrationエンドポイントは事前に共有(out-of-band)
    - registration access tokenをStep1で出してもいい　short lived & usable only once
  - Step2 : Discovery and openid Configuration
    - issuerのURLに付加する形が必要
  - Step3 : Client Registration
    - Authorizationヘッダーで制御していい
    - RegisterをPOSTする前にToolは正しい宛先か判断する(must) product_family_code&version
    - deployment_idをClientと紐づけている場合は、自動発行＆Responceしてよい
  - Step4 : Registration Completed and Activation

OIDC Discovery、OIDC Dynamic Client Registration　概要

OpenID Connect Discovery

できること	OPの構成情報を自動的に発見することができる
対応済みOP	Google, MS, Apple, Facebook, Yahoo! JP等々
MUST要件	Well-Knownエンドポイントの公開(issuer, authorization_endpoint)
その他	各種OIDCのサポートしている項目や、OPのポリシー等々のメタ情報がWell-Known EPで提供

OpenID Connect Dynamic Client Registoration

できること	RPの情報をOPに登録することができる
対応済みOP	(代表的なもの)Okta, OneLogin
MUST(ミニマム)要件	Req.時 redirect_urls、Res.時 client_id
その他	デプロイパターンが複数取りえる　後述

https://zenn.dev/nhosoya/articles/c5a897b9b1974ae4ada6
https://developer.okta.com/docs/reference/api/oauth-clients/
https://developers.onelogin.com/openid-connect/api/dynamic-client-registration

OpenID Connect Dynamic Client Registration①

OIDC DCR自体だけで登録が行えるわけではない(システム全体を定義していない)
登録が行うことができるReq./Res.について定義している
★登録APIの保護のために必要なトークンをどのように取得するかについては規定していない

3. Client Registration Endpoint（引用）

The Client Registration Endpoint is an OAuth 2.0 Protected Resource through which a new Client registration can be requested. **The OpenID Provider MAY require an Initial Access Token that is provisioned out-of-band (in a manner that is out of scope for this specification) to restrict registration requests to only authorized Clients or developers.**

CDRのデプロイパターン(Curityにて)
①Open Registoration　　:フルオープン　誰でもRP登録できちゃう　普通やらない
②Client Authenticated Registration　　:Client Credential flowで得られたトークンを持つクライアントで許可される
③User Authenticated Registration　　:ユーザーの認証情報(トークン)で許可　クライアントは識別のみ
④No Registration:　　:無効化　デフォルト

②ユーザーがセルフサインアップができない場合（RP側にユーザー作れずに、でもSSOしたいケース？）
　①に対するスピードバンプになっている　最初に暫定で使うなどはある
　Client Credential flowで事前に共有した情報などで、許可するクライアントを判断する

③アプリを広く配布する際に通常利用される　アプリ内のユーザーが認証されたことで、許可を判断する

OpenID Connect Dynamic Client Registration②

OIDC DCR自体だけで登録が行えるわけではない(システム全体を定義していない)
登録が行うことができるReq./Res.について定義している
★クライアントが自身を登録するために必要なトークンをどのように取得するかについては規定していない

★OIDC DCRでRPの登録と、Client ID(や各種情報)の取得はできるが、エンドポイント自体の保護は別途行うのが一般的そう

Registerエンドポイントの保護　実際
・Okta　:Okta APIトークンをAdmin Consoleで発行する　同トークンをヘッダーにSSWSで送付
・OneLogin　:AdminアカウントでAPI Credentialをアクセストークンを発行(権限を設定したtoken)
API Client Credentialで取得したトークンでも可能

https://developer.okta.com/docs/reference/api/oauth-clients/
https://help.okta.com/ja-jp/Content/Topics/Security/API.htm
https://developers.onelogin.com/api-docs/2/oauth20-tokens/generate-tokens-2
https://developers.onelogin.com/api-docs/1/getting-started/working-with-api-credentials
https://developers.onelogin.com/openid-connect/api/client-credentials-grant

LTI Dynamic Registration概要

- LTI PlatformへのLTI Toolの登録を簡易にするための標準仕様
- OpenID Connect Discovery、OpenID Connect Dynamic Client Registration、[RFC7591] OAuth 2.0 Dynamic Client Registration Protocolの仕様を活用
- OpenID Connect Discovery、OpenID Connect Dynamic Client Registrationと異なる点
  - Platformからトリガーされる点
  - Tool, Platform間で長く関係が続く
  - Platform側でToolを使うか判断する（DCRも同じでは？）
- Metadata Definitions
  - OIDC+LTI連携で必要な情報の記述が可能⇒Platform and Tool Configuration Metadata Definitions
  - カスタムフィールドも持てる
- Registration Protocol
  - Step1 : Registration Initiation Request
    - ToolのRegistrationエンドポイントは事前に共有(out-of-band)
    - registration access tokenをStep1で出してもいい　short lived & usable only once
  - Step2 : Discovery and openid Configuration
    - issuerのURLに付加する形が必要
  - Step3 : Client Registration
    - Authorizationヘッダーで制御していい
    - RegisterをPOSTする前にToolは正しい宛先か判断する(must) product_family_code&version
    - deployment_idをClientと紐づけている場合は、自動発行＆Responceしてよい
  - Step4 : Registration Completed and Activation

Metadata Definitions

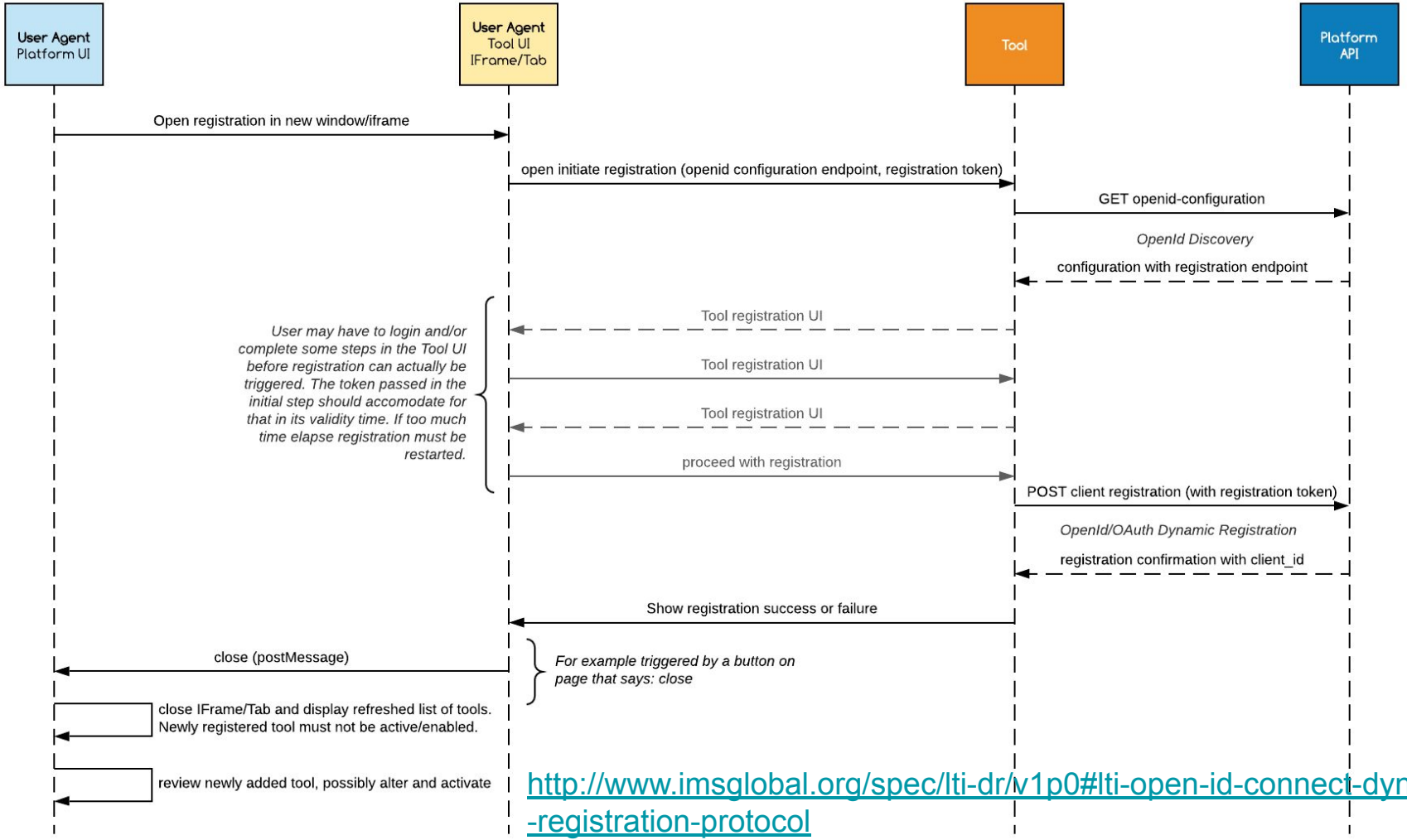
Property	Usage for LTI Recommendation
issuer	Platform's issuer value. As per IMS Security Framework and LTI Specification, the Issuer Identifier is a case-sensitive URL, using the HTTPS scheme, that contains scheme, host, and optionally, port number, and path components, and no query or fragment components.
authorization_endpoint	URL of the OAuth 2.0 Authorization Endpoint.
registration_endpoint	URL of the registration endpoint; may be a one time use only end-point and/or protected by access token.
jwks_uri	URL of the Platform JWK Set endpoint; may be specific per registration if the platform's issued a dedicated discovery end-point for that registration.
token_endpoint	URL of the endpoint for the tool to request a token to access LTI (and possibly other) services.
token_endpoint_auth_methods_supported	Must contain `private_key_jwt`; may offer additional values.
token_endpoint_auth_signing_alg_values_supported	Must contain `RS256`; may offer additional values.

https://purl.imsglobal.org/spec/lti-platform-configuration	A JSON Object object containing LTI specific configuration details for this registration. See below.

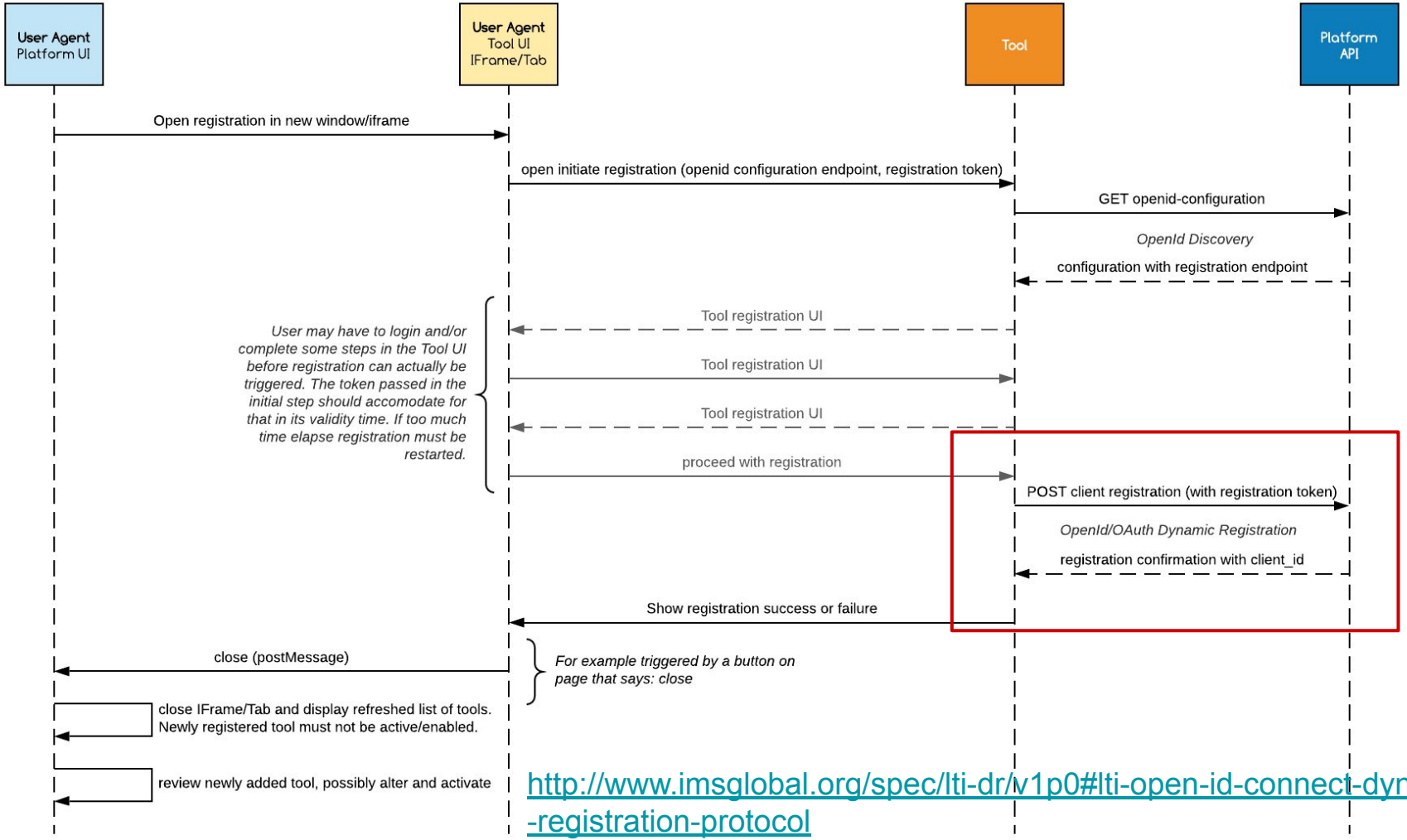
Property	Definition
product_family_code	Product identifier for the platform.
version	Version of the software running the platform.
messages_supported	An array of all supported LTI message types. See below for message supported properties.
variables (optional)	An array of all variables supported for use as substitution parameters.

http://www.imsglobal.org/spec/lti-dr/v1p0#platform-and-tool-configuration-metadata-definitions

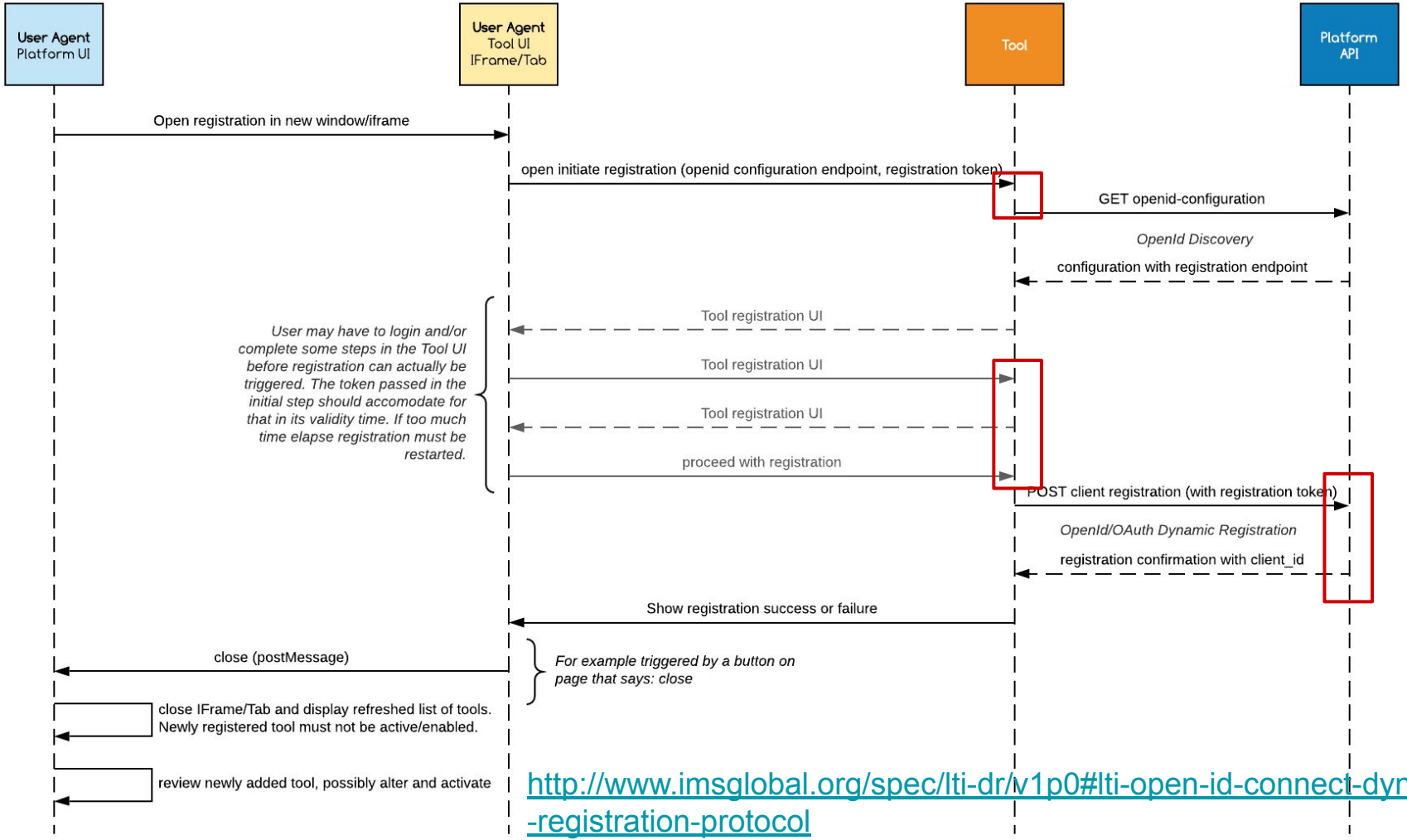
```
{
    "issuer": "https://server.example.com",
    "authorization_endpoint":  "https://server.example.com/connect/authorize",
    "token_endpoint": "https://server.example.com/connect/token",
    "token_endpoint_auth_methods_supported": ["private_key_jwt"],
    "token_endpoint_auth_signing_alg_values_supported": ["RS256"],
    "jwks_uri": "https://server.example.com/jwks.json",
    "registration_endpoint": "https://server.example.com/connect/register",
    "scopes_supported": ["openid", "https://purl.imsglobal.org/spec/lti-gs/scope/contextgroup.readonly",
       "https://purl.imsglobal.org/spec/lti-ags/scope/lineitem",
       "https://purl.imsglobal.org/spec/lti-ags/scope/result.readonly",
       "https://purl.imsglobal.org/spec/lti-ags/scope/score",
       "https://purl.imsglobal.org/spec/lti-reg/scope/registration"],
    "response_types_supported": ["id_token"],
    "subject_types_supported": ["public", "pairwise"],
    "id_token_signing_alg_values_supported":
      ["RS256", "ES256"],
    "claims_supported":
      ["sub", "iss", "name", "given_name", "family_name", "nickname", "picture", "email", "locale"],
     "https://purl.imsglobal.org/spec/lti-platform-configuration": {
        "product_family_code": "ExampleLMS",
        "messages_supported": [
            {"type": "LtiResourceLinkRequest"},
            {"type": "LtiDeepLinkingRequest"}],
        "variables": ["CourseSection.timeFrame.end", "CourseSection.timeFrame.begin", "Context.id.history",
"ResourceLink.id.history"]
    }
}
```



http://www.imsglobal.org/spec/lti-dr/v1p0#lti-open-id-connect-dynamic-registration-protocol



http://www.imsglobal.org/spec/lti-dr/v1p0#lti-open-id-connect-dynamic-registration-protocol



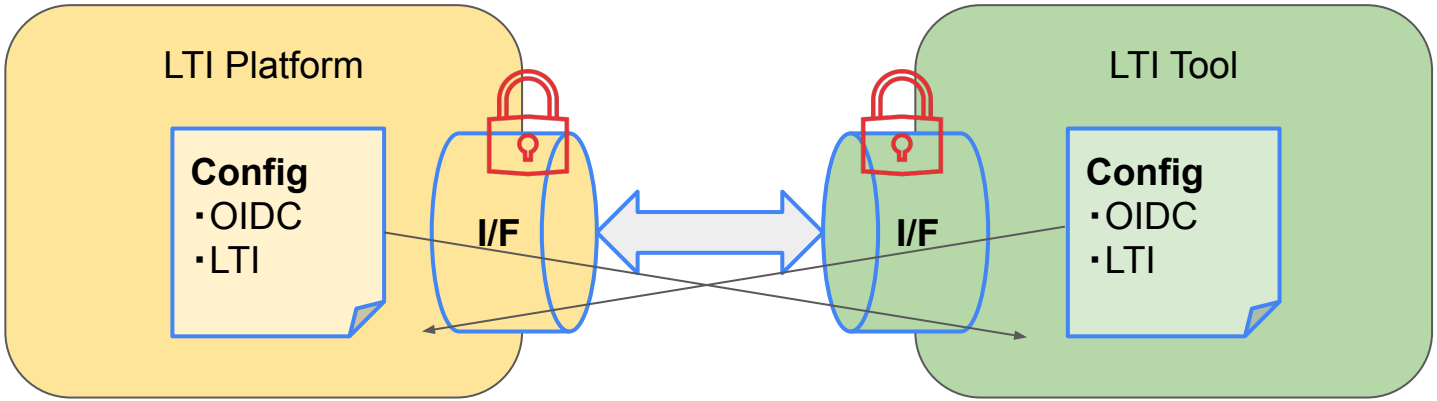
http://www.imsglobal.org/spec/lti-dr/v1p0#lti-open-id-connect-dynamic-registration-protocol

LTI DR仕様　イメージと検討事項

LTI DR仕様に関して標準化検討事項
①Platform/Tool Configrationの記載内容検討　※OneRosterと同様イメージ
②I/Fの保護、エンドポイントのデプロイパターンの標準化
※I/F Protocolに関しては基本標準仕様の通り　一部決めが必要なのかは今後

所感
①はそこまで手を加えない？　揺らぎとなりOptionalを定義することでプラスな部分を規定する
②は学習eポータル標準側での定義？　日本1 Edtech として規定するスコープ外？

青枠:LTI DR規定
赤枠:LTI DR規定外



LTI DR所感

- LTI DRで連携の初期設定が楽になるパターンは、いくつか公開情報を皆見れる環境に置いておくことで可能
- 以下を公開＆仕様化しI/Fのデプロイパターン,保護が標準化できればLTI DRが可能
  公開
  Platform ：Issure URL
  Tool ：JWKs, LTI DR init Endpoint
  仕様化
  ・Register エンドポイントのデプロイパターン
  保護にOAuth 2.0 Client Credential使う？
  ・(JWKsの運用目安）
- 各Platform/ToolがLTI DRに登録することで、手作業が減るところはある
- ★まだ実装/デプロイを具体イメージできる理解ではない

今後の個人イメージ

- ユースケースの検討や取り込む価値の判断
- 平行してLTI DRとしてどこまで何ができるのか検証

ディスカッションの案

- LTI連携に関して、Dynamic Registrationを取り込むことで効果がでるか？
  - ユースケースの想定
  - 代替案の有無と比較
- 代替案
  LTI連携情報などGithub Private リポジトリで共有
  管理,書き込みなどの権限制御が可能　監査なども可能
- 疑問点（今後検証などしてクリアにしてくべき箇所）
  影響度など加味して
- 実証できるか？

疑問点

カテゴリ	内容	
仕様上	1EdTech Security Frameworkとの関係性は？	
実装上	iframeでの実施時のCSPヘッダーの設定等	
実装上	Client_idの生成タイミング	
実運用上	情報更新あった場合（Registration Updateのみでは対応できなそう？）	
実運用上	Issureを顧客ごとに分けているPlatform/Toolはどうなる？	