

AIアプリケーション向け 軌道上実証環境概要

2025 年 7 月 15 日 初版

宇宙航空研究開発機構

1. はじめに

本資料は、D-OBEC の軌道上実証を目指すアプリ開発者向け軌道上実証環境について定めるものである。

2. 用語

本資料で使用する用語を以下に規定する。

表 2-1 用語一覧

No.	用語	説明
1	D-OBEC	<u>D</u> emonstration of <u>O</u> n- <u>B</u> oard <u>E</u> dge <u>C</u> omputing(軌道上エッジコンピューティング技術実証)の略。名前の通り、軌道上でエッジコンピューティング技術を実証するためのプロジェクト名称。
2	Docker	コンテナ仮想化技術を用いてアプリケーションを開発・配置・実行するためのオープンプラットフォーム。
3	Dockerfile	Docker コンテナイメージを生成するための生成条件を定義するテキストファイル。コンテナアプリの元となる Docker コンテナイメージ(BaseImage)や設定情報等を定義する。
6	コンテナアプリ	D-OBEC の軌道上実証に供する(候補を含む)、アプリ開発者が開発した Docker コンテナイメージ。JAXA から提示する BaseImage をベースイメージとして(BaseImage を Dockerfile の FEOM 句に指定して)作成する。
7	アプリ開発者	コンテナアプリの開発者。
8	アプリ管理者	コンテナアプリの管理者(JAXA)。アプリ開発者から提供されるコンテナアプリの事前検証、軌道上環境へのデプロイを行う。
9	BaseImage	VPUドライバ、AI系ライブラリ、gRPC等を含む Docker コンテナイメージ。コンテナアプリの開発環境または実行環境として JAXA からアプリ開発者へ提供する。また、コンテナアプリの共通部分として衛星実機に搭載する。
11	QPS-SAR	QPS 研究所が開発したレーダ衛星で、高分解能の X バンド SAR を搭載する。コンテナアプリを軌道上実証するためのホスト衛星となる。

3. 軌道上実証環境

3.1. 軌道上計算機環境

コンテナアプリを動作させる軌道上ホスト環境を、表 3-1 に示す。

表 3-1 ホスト動作環境

項目	説明	
プロセッサ	AMD Ryzen™ Embedded V1605B with Radeon™ Vega 8 Graphics	
RAM	24 GB DDR4 ECC	
ストレージ	SSD: 4TB(うち、ユーザ領域 1TB) ※ユーザ領域は全てのコンテナアプリで共有	
HW アクセラレータ	Intel Movidius Myriad(NCS2 相当 VPU) 1Unit	
OS	Ubuntu 20.04.6 相当	
機械学習 SDK	VPU 用	Open VINO 2022.3.1, Torch, Tensorflow, Keras, onnx, opencv, scikit-learn, scipy 等
	CPU 用	Torch, Tensorflow, Keras, onnx, opencv, scikit-learn, scipy 等
コンテナエンジン	docker CE 24.0.9 containerd 1.6.28 docker-compose-plugin 2.27.1 docker-buildx-plugin 0.13.1	

3.2. BaseImage の種類

アプリ開発者がコンテナアプリを作成する際にベースイメージとして使用する(Dockerfile の FROM 句に指定してビルドする)2 種類の BaseImage を提供する。

- 1) AMD の CPU を使用する AI アプリ向けの環境 (※GPU は、本コンテナアプリでは利用できない)
- 2) VPU を使用した推論処理を行う AI アプリ向けの環境

3.3. 入力データ

QPS-SAR の X バンド SAR 画像。なお、1 シーン分の衛星画像ファイルのサイズは、最大 4GB である(画像形式や撮像条件に依存する)。

(1) データフォーマット

- ・ SLA(ピクセル毎の値を float32(4byte)で表したもの)
- ・ SLC(ピクセル毎の複素数値(実部と虚部)の値を float32(4byte)で表したもの)
- ・ JPEG

3.4. 動作制約

3.4.1. コンテナアプリのサイズ

コンテナアプリのサイズを 30MB 以下とすること(このサイズには、学習済みモデルファイルや追加でインストールした独自ライブラリ等、コンテナアプリが動作する際に必要なファイルが全て含まれる)。

3.4.2. コンテナアプリの数

1 つのユースケースに対し、原則 1 つのコンテナアプリとして実装すること。複数のコンテナアプリを同時に起動して動作させることを禁止する。

3.4.3. コンテナアプリの動作時間

コンテナアプリは、CPU 割当やメモリ制限を課した状態で、15 分以内(軌道上における計算機の性能評価結果を踏まえて最終的に時間を判断するため多少増減する可能性あり)に処理を完了すること。

なお、コンテナアプリが利用可能な CPU やメモリは、それぞれ CPU1 コア(2 スレッド)、メモリ 6GB を上限とする(コンテナアプリ実行条件ファイルに定義されたリソース制限値を変更しないこと)。

3.4.4. コンテナアプリの出力データサイズ

衛星のダウンリンク回線レートを考慮し、コンテナアプリの出力データサイズ(全出力データの合計)は、1 回の実行あたり 250MB 以下とすること。

なお、入力データのうち衛星画像については、衛星によってダウンリンクされるため、コンテナアプリ側で出力データに衛星画像をコピーする必要はない(コンテナアプリによって加工した衛星画像はこの限りではない)。

3.4.5. コンテナアプリのログ

コンテナアプリのログ情報は、出力データの一つとしてファイル出力すること(標準出力および標準エラー出力のみの場合、地上で確認できないため)。なお、同ファイルも出力データサイズに内包される点に留意のこと。

3.4.6. コンテナアプリの実行フロー

軌道上実証におけるコンテナアプリの実行フローは、図 3-4 の通りとなる。

このときのコンテナアプリの実行可能時間(動作時間)は、計算機の稼働時間から前処理と後処理に必要な時間を差し引いて、1 実行あたり最長 15 分(軌道上における計算機の性能評価結果を踏まえて最終的に時間を判断するため多少増減する可能性あり)までとなる。

コンテナアプリは、同時間内に処理を終えることを基本とし、それが難しい場合は、適宜ハンドオーバー機能(前回中断した処理を再開する機能の意)を具備することにより対応すること(1 ユーザあたりの実証機会は、4 回程度の予定)。

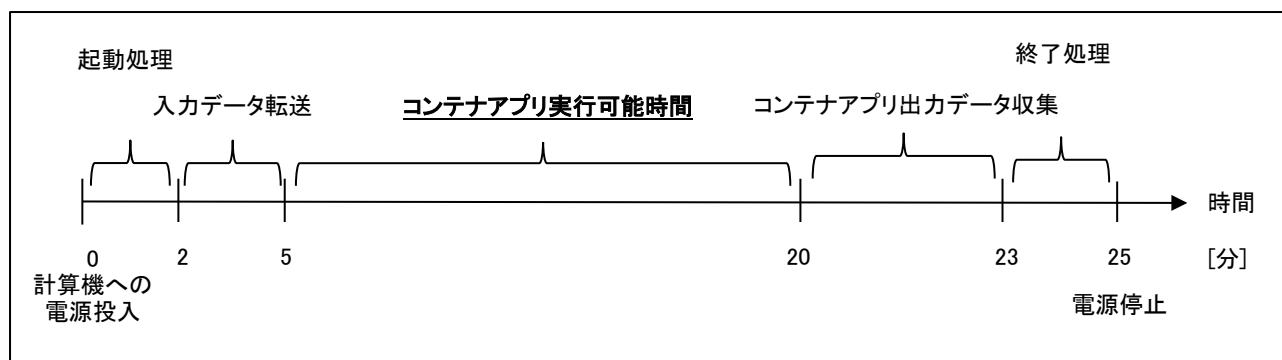


図 3-4 コンテナアプリの実行フロー

4. コンテナアプリ作成支援

4.1. VPU スティックの貸し出し

VPU 環境を持たないアプリ開発者(希望者)を対象として、JAXA が保有する VPU スティック(Intel NCS2)の貸し出しを行うことを計画中。なお、数に限りがあるため、貸出数はアプリ開発者の募集単位ごとに 1 デバイスを上限とする。

4.2. コンテナイメージ差分作成ツールの提供

衛星のアップリンク回線レートに鑑みると、BaseImage を含むコンテナイメージをそのまま衛星へデプロイ(アップリンク)することは難しい。そのため、軌道上動作環境に予め登録済みのレイヤ(=BaseImage)はデプロイせず、それ(BaseImage)との差分レイヤのみを衛星へデプロイする方針を採る。

アプリ管理者に対しては、BaseImage との差分レイヤを取り出して、イメージファイルを作成するツールを提供する。

5. 軌道上実証の流れ

事前検証

- ・ アプリ管理者は、アプリ開発者から提出された提出物をもとに、コンテナアプリ等をレジストリに登録し、脆弱性診断を行う。
- ・ アプリ管理者は、事前検証環境でコンテナアプリを実行し、CPU/メモリ/ストレージの使用量、消費電力、温度、動作時間、出力データサイズ、各種ログ等を確認し、軌道上実証が行えるかどうかを確認する。

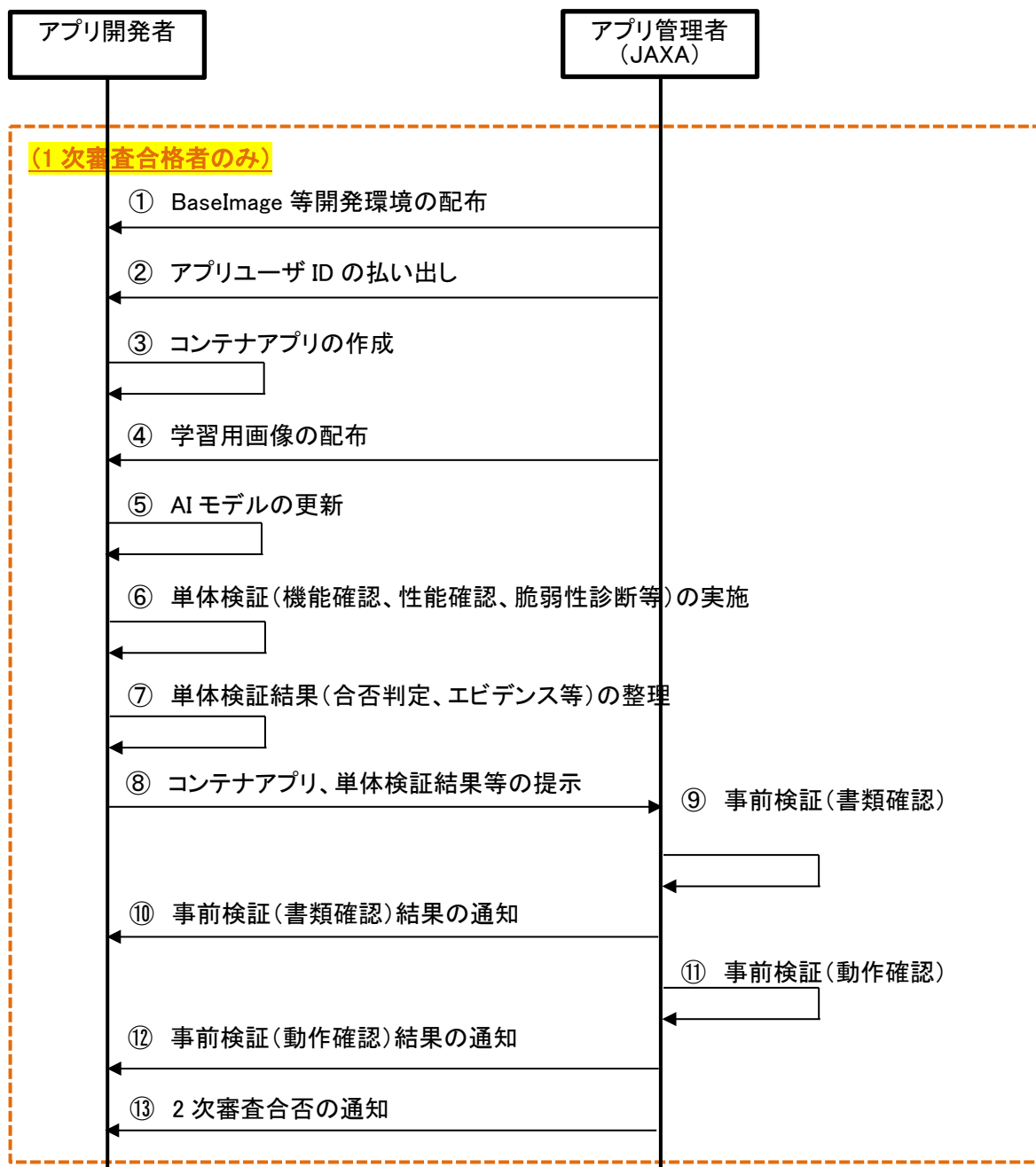


図 3-1 軌道上実証の流れ(1/2)

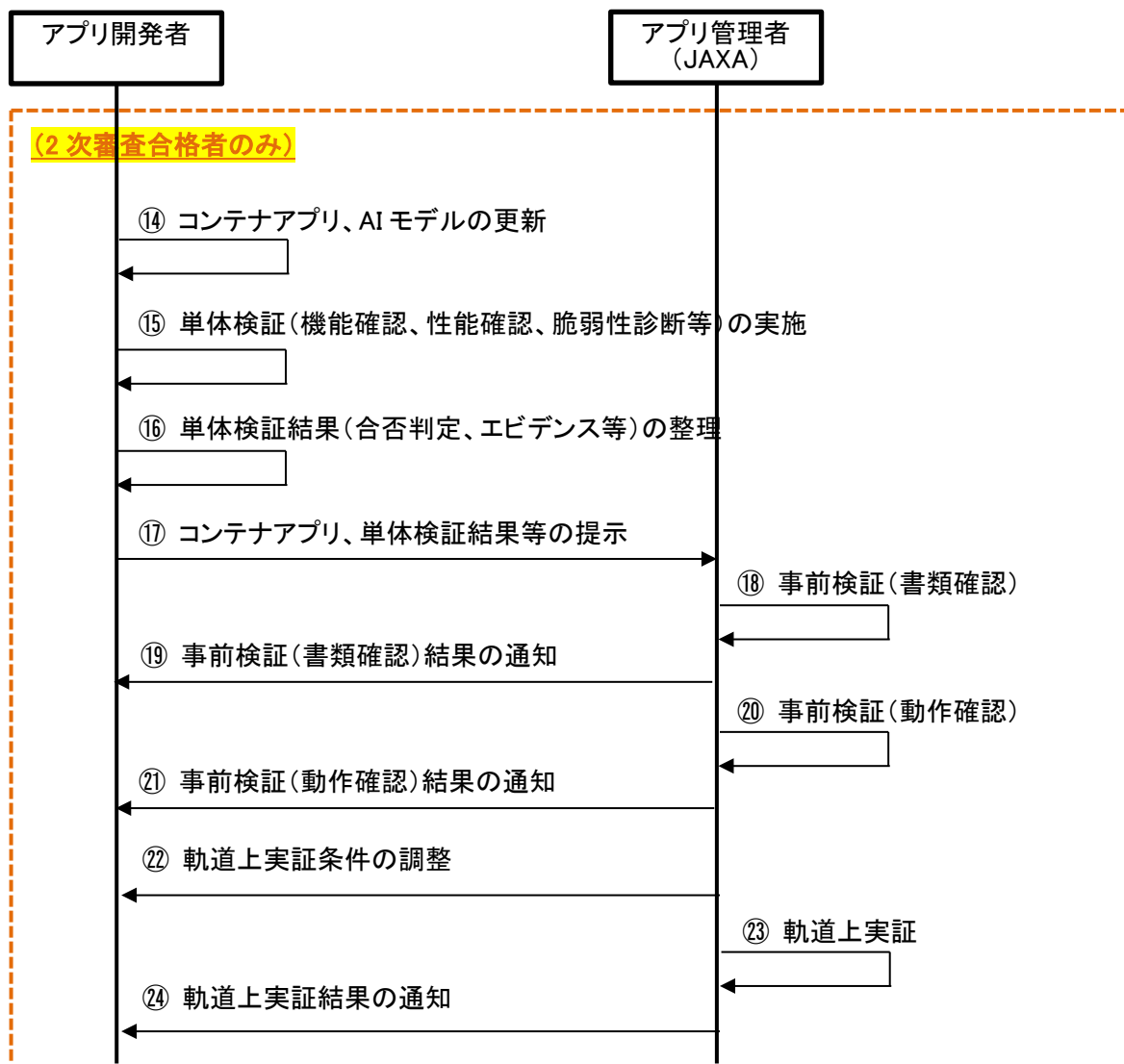


図 5-1 軌道上実証の流れ(2/2)