Smart IoT Update for Yocto based Azure IoT devices

The following document explores and articulates the design specification for implementing upgrade of Yocto Linux Image based Azure IoT Devices. For additional questions about this spec, scenario or product area, please consult the aliases listed on the authors line above.

# Overview

## Scenario Description

Azure IoT devices need to be updated every now and then to ensure that they are always running the latest secure silicon and OS binaries.

Currently in Yocto linux OS based Azure IoT devices, two alternatives are available for performing updates: a full-image-swap update and a package update. There are different challenges associated with both of these options.

With full-image-swap update, images are downloaded over network which slows down the network by a large extent. On the other hand, with the package update, full scale updates are not feasible which is occasionally required in case of updating the OS.

With differential update model, there will be a substantial decrease in network congestion, and it will help in improving the IoT OEM and ISV experience.

## Target User or System

This Design spec targets the Yocto Linux image-based Azure IoT devices. Majorly enterprise customers for Azure IoT products are the target audience here.

## Problem Statement and Supporting Customer Insights

Below are some of the major challenges with the existing ADU E2E workflow.

### Massive Download Size

With the current OS Swap implementation of installing the updates, Azure IoT device needs to download the entire Yocto Linux image. Most of the times only few packages or files are changed as part of the update but still entire image needs to be downloaded over the network.

### Scalability

With the current huge download size of Yocto Linux image and the corresponding network impact, customers are struggling with applying updates to just a few handful of devices. Even though updates to the current OS are minor, they need to download the whole image.

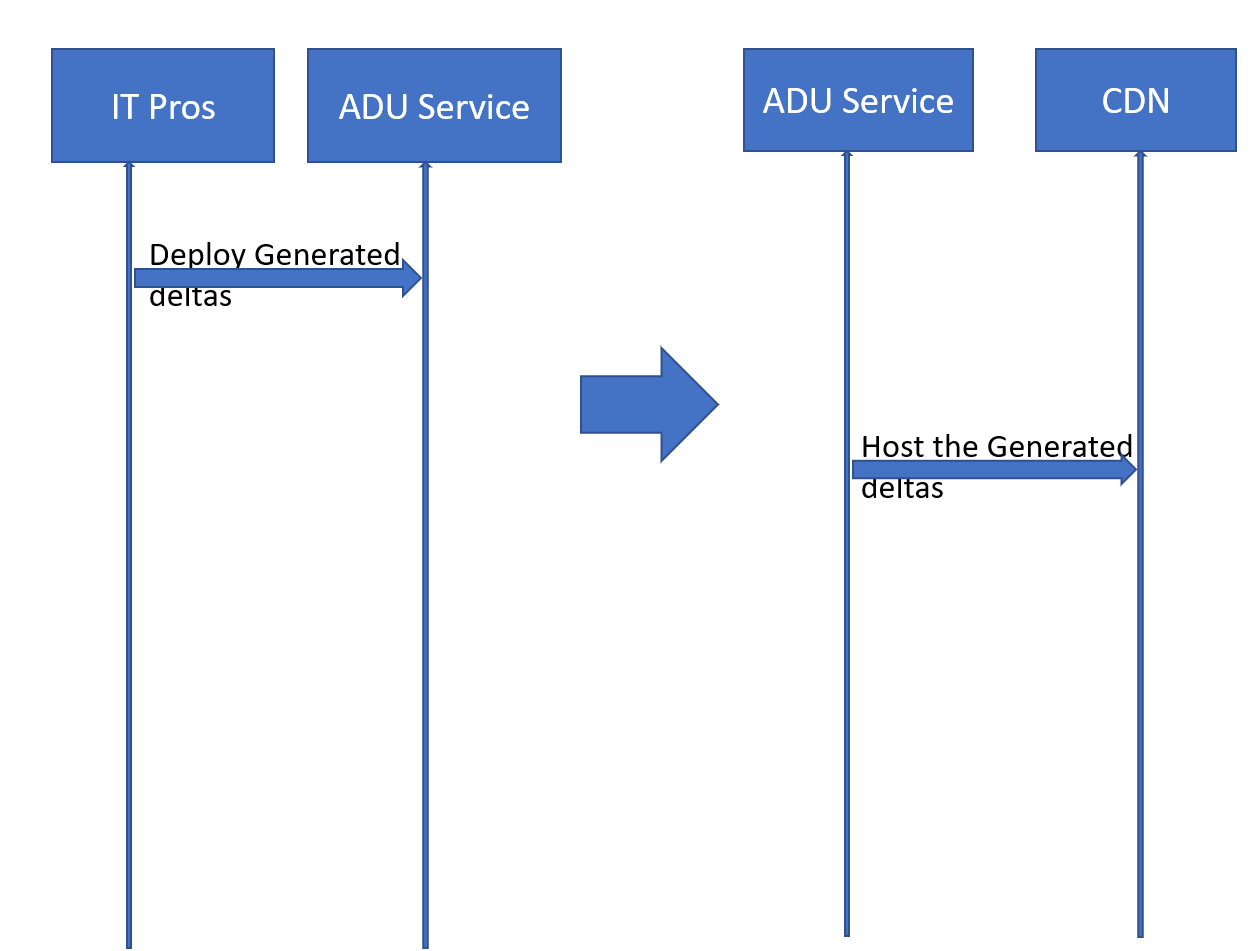
# Detailed Design

## High level workflow

### Deltas Generation and Validation

### 

### Uploading the Generated Deltas



### 

### Deploying the Deltas via ADU

### 