

CFP Overview

- [Mission](#)
- [Structure](#)
- [Properties in Scope](#)
- [Objectives](#)
- [Guiding Principles](#)

Mission

The Customer Facing Platform (CFP) engineering team provides **front-end tools, libraries, and components** that our digital experience engineering teams use to create a world-class user experience for our customers.

We abstract complexity for web experience codebases and provide standardized solutions - a platform - so that teams can focus on delivering features that matter to our customers and business stakeholders.

Structure

CFP is comprised of two main elements: Application Development Platform, and UI Platform

- [Application Development Platform](#) (microfrontends, or MFEs) is a collection of standard libraries and tools that make it simple for everyone to create, develop, test and deliver applications that can be deployed on customer facing properties.
- [UI Platform](#) (Digital HIG) provides the engineering implementation of the Digital HIG design system and other tools used to reproduce design patterns in applications.

Properties in Scope

[Autodesk.com](#)

Autodesk Account (& Profile)

Autodesk eStore

Autodesk Knowledge Network

Autodesk Forums

Objectives

Improve Customer Experience

Customers traverse multiple sites during user journeys, each with its own features, look & feel, and user context.

- How can we provide customers with the capabilities they need wherever they are?
- How can we personalize customers' progress through their journeys?
- How can we make the user experience more consistent and predictable?

Enable Internal Self Service

Product management, experience design, web publishing, research and marketing follow coordinated workflows and work closely with site developers to author content, integrate features on sites and run experiments.

- How can we give partners greater independence and autonomy?
- How can we enable partners to quickly and nimbly transform the customer experience?
- What tools do they need to do so?

Increase Developer Productivity

Developer teams work within a single site or domain silo building location-specific features and generic APIs with technologies largely unique to it.

- How can developers work efficiently across sites or domains?
- How can we develop and deliver site-agnostic capabilities?
- How do we build **shared** landmark services, infrastructure and Customer Experience tools?

Guiding Principles

Enable partner self-service

- Reduce dependency on engineering for publishing, personalization, experimentation

Personalize based on first-party data

- Know who I am and what I have, across the whole journey

Build capabilities, not destinations

- Domain-driven design. Each capability owned in one place by the right team and deployed in many destinations

Decouple Content from presentation

- Structured, rather than formatted, content as a service. Common taxonomy applied to content. Content not duplicated across systems

Invest in an intentional developer ecosystem

- Unified cross-org design system. One way to do things: landmark services, tooling, and front-end development
- Improved developer efficiency and interchangeable teams

Thoughtfully balance autonomy against simplification of our technologies

- Evaluate which technologies are best in class and actively promote them for adoption
- Reduce required coordination by decoupling capabilities wherever possible