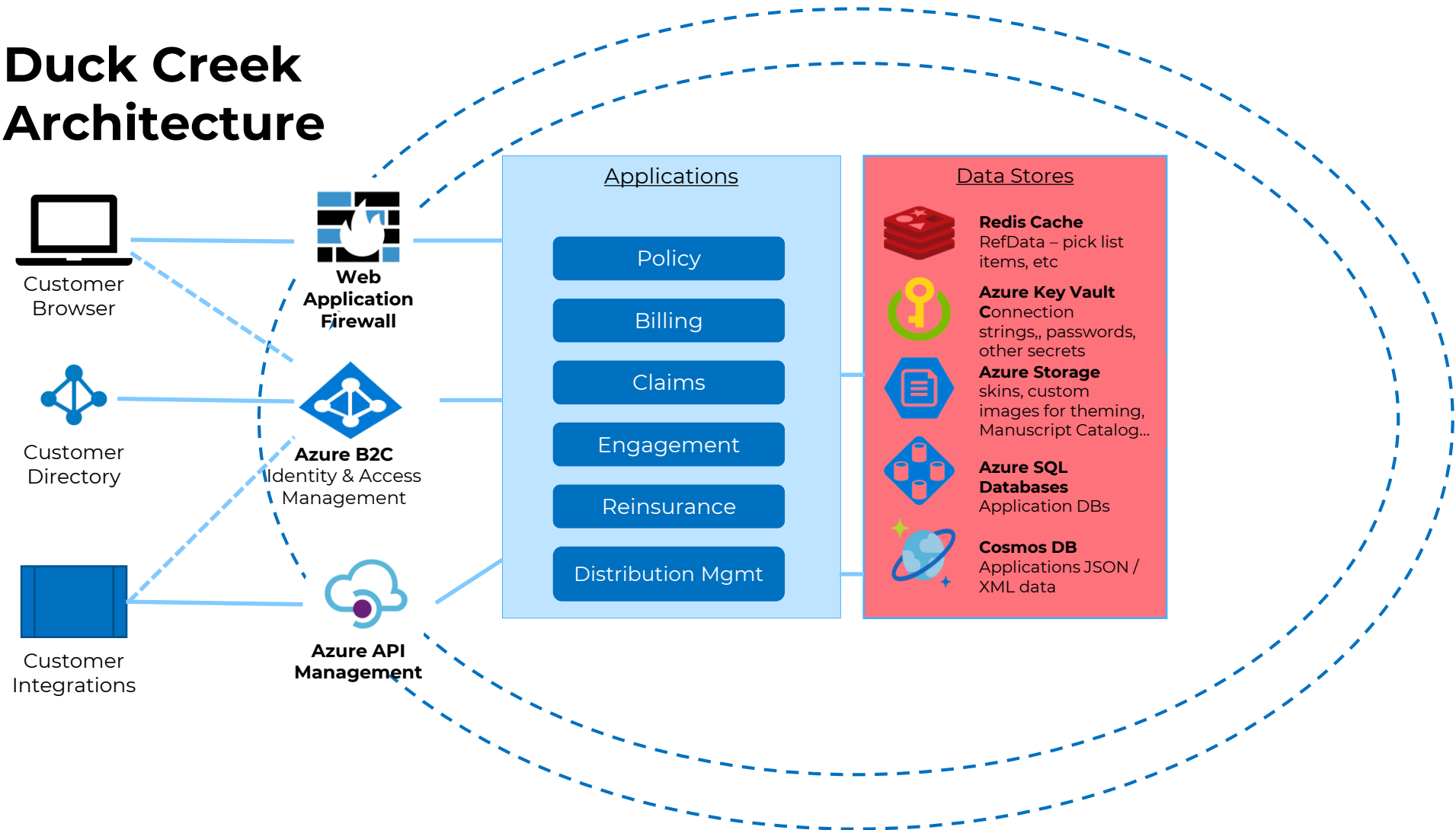




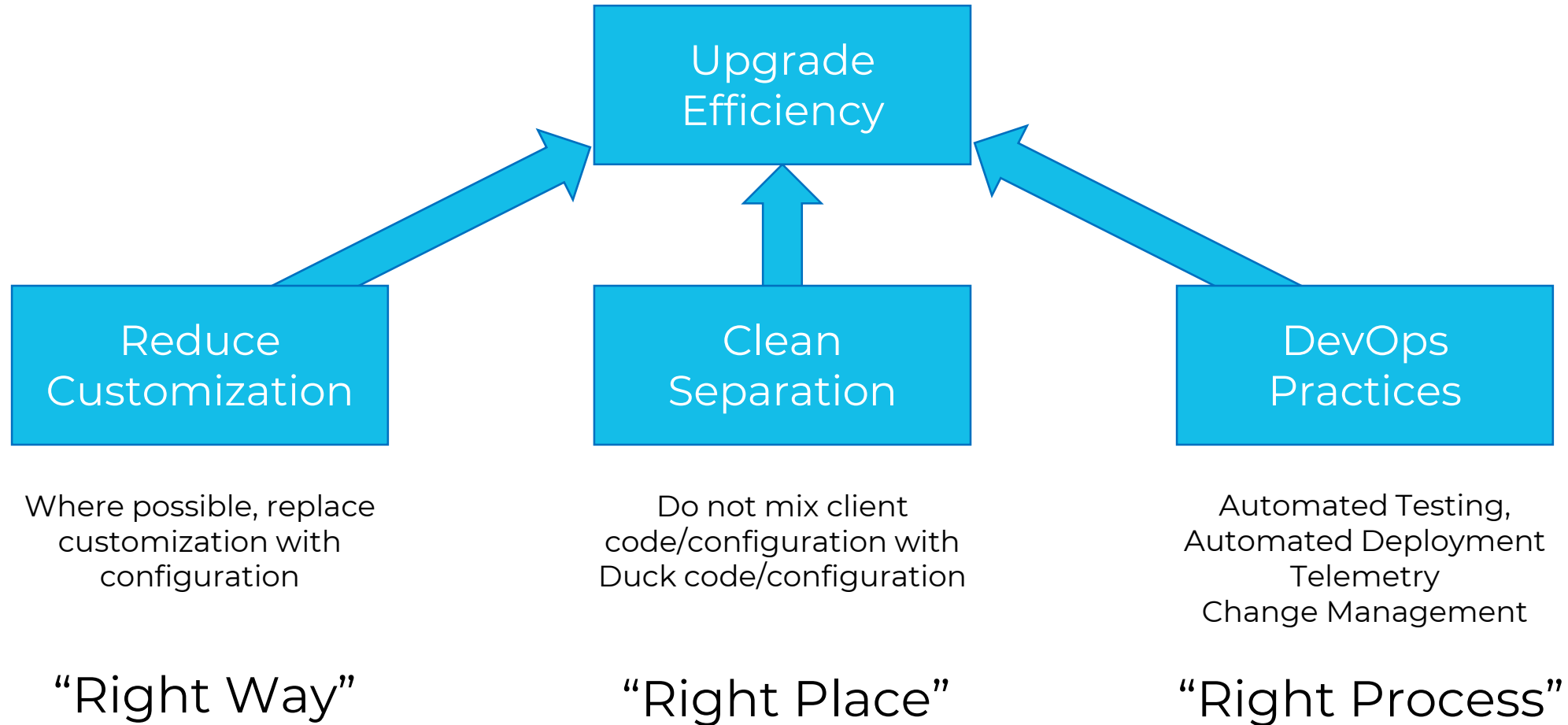
Duck Creek  
Technologies

# Duck Creek Architecture Preparedness

# Duck Creek Architecture



# Implementation Guidelines

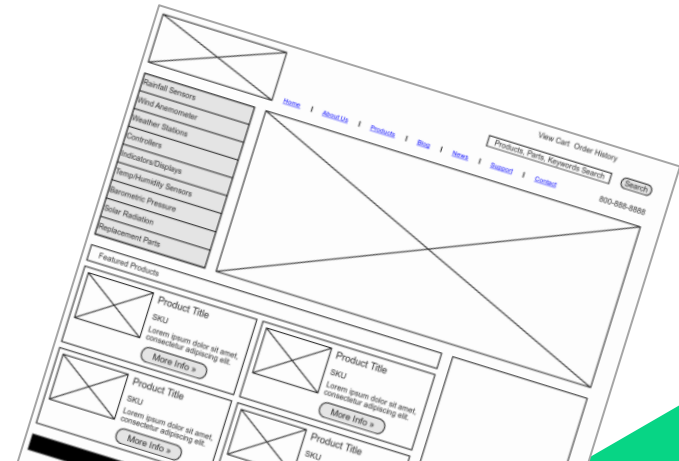


# UI Implementation Guidelines

- ▶ Use new Page Builder based page definition for Express
  - No more custom system pages
  - No custom Express post actions
  - No updates/extensions to core JavaScript
  - No Skin changes
  - Only variable-based CSS theme updates
- ▶ Use upgrade best-practices in Claims and Party Studio
  - No custom pages
  - No updates/extensions to core JavaScript
  - No layout or behavior altering CSS updates

## Alternatives?

1. Use existing pages, actions, CSS, etc.
2. Inquire about base application enhancement
3. Utilize headless approach



# Custom Logic Implementation Guidelines

- ▶ No custom code components
  - No Custom Business Objects (CBOs)
  - No ManuScript Functions
  - No custom Activity objects
  - No custom Expression Functions
  - No core JavaScript code changes

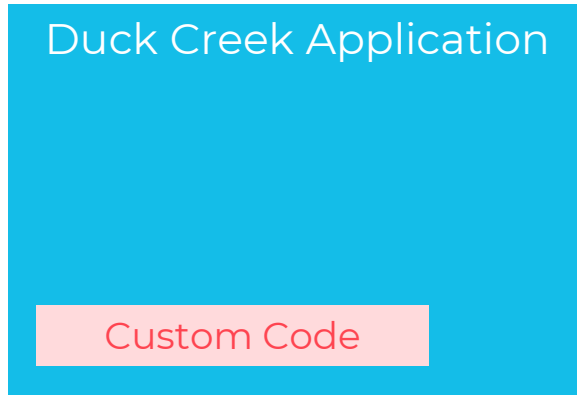


## Alternatives?

1. Use existing API and/or configuration (consider AMIs)
2. Externalize as a microservice
3. Inquire about base application enhancement

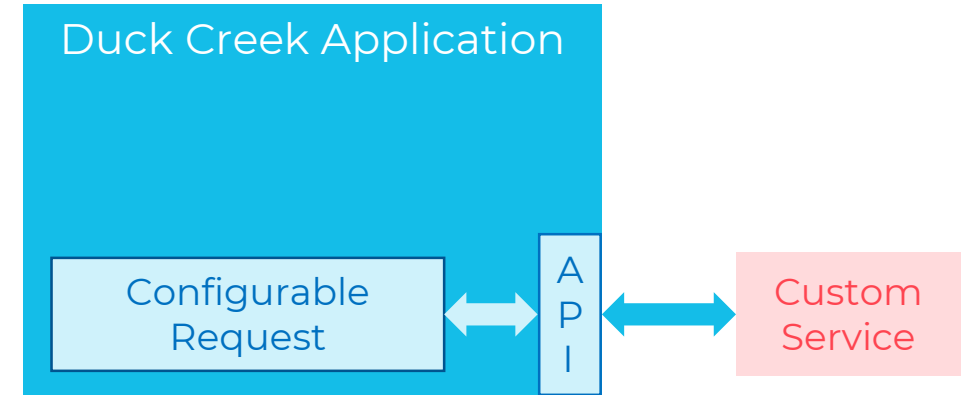
# Clean Isolation of Custom Code

## Existing Customization Pattern



- ▶ Customizations are compiled and deployed alongside the Duck Creek components
- ▶ This can introduce build dependencies, leading to conflicts at time of upgrade

## Improved Customization Pattern



- ▶ Customizations are externalized as stand-alone services, integrated via APIs at **run time**
- ▶ Standard features are used to configure requests to the external services
- ▶ The request configurations and APIs are maintained backward-compatible across releases (or feature-flagged)

**Result: Eliminate all build conflicts resulting from upgrades**

# Custom Business Object Alternatives

Don't Do	Best Practice	How this changes
CBO for Integration	<ul style="list-style-type: none"><li>a) Call an external service using the <a href="#">Send.httpRq</a> or <a href="#">Send.postRq</a> feature</li><li>b) If additional functionality is needed, like negotiating authentication tokens, caching, or feeding a data lake, implement those as an external service or leverage something like Azure API manager to add that functionality.</li></ul>	Strictly enforced
CBO for custom business logic	<ul style="list-style-type: none"><li>a) Move logic to Manuscript, or AFS config</li><li>b) Use an AMI</li><li>c) Move logic to external service, as per above for Integration</li><li>d) Call in to Anywhere APIs from external service if required</li><li>e) Request a new feature</li></ul>	Strictly enforced

Implication: All custom business objects converted to configurations or externalized.

# Database Implementation Guidelines

- ▶ No changes to database schema
- ▶ No changes to database functions/procedures
- ▶ No direct database connections

## Alternatives?

1. Use existing capabilities
  - Application API
  - Query API provided for basic access to schema
  - Utilize Data Insights
2. Externalize to separate external database
3. Inquire about a base application enhancement





# DevOps Implementation Guidelines

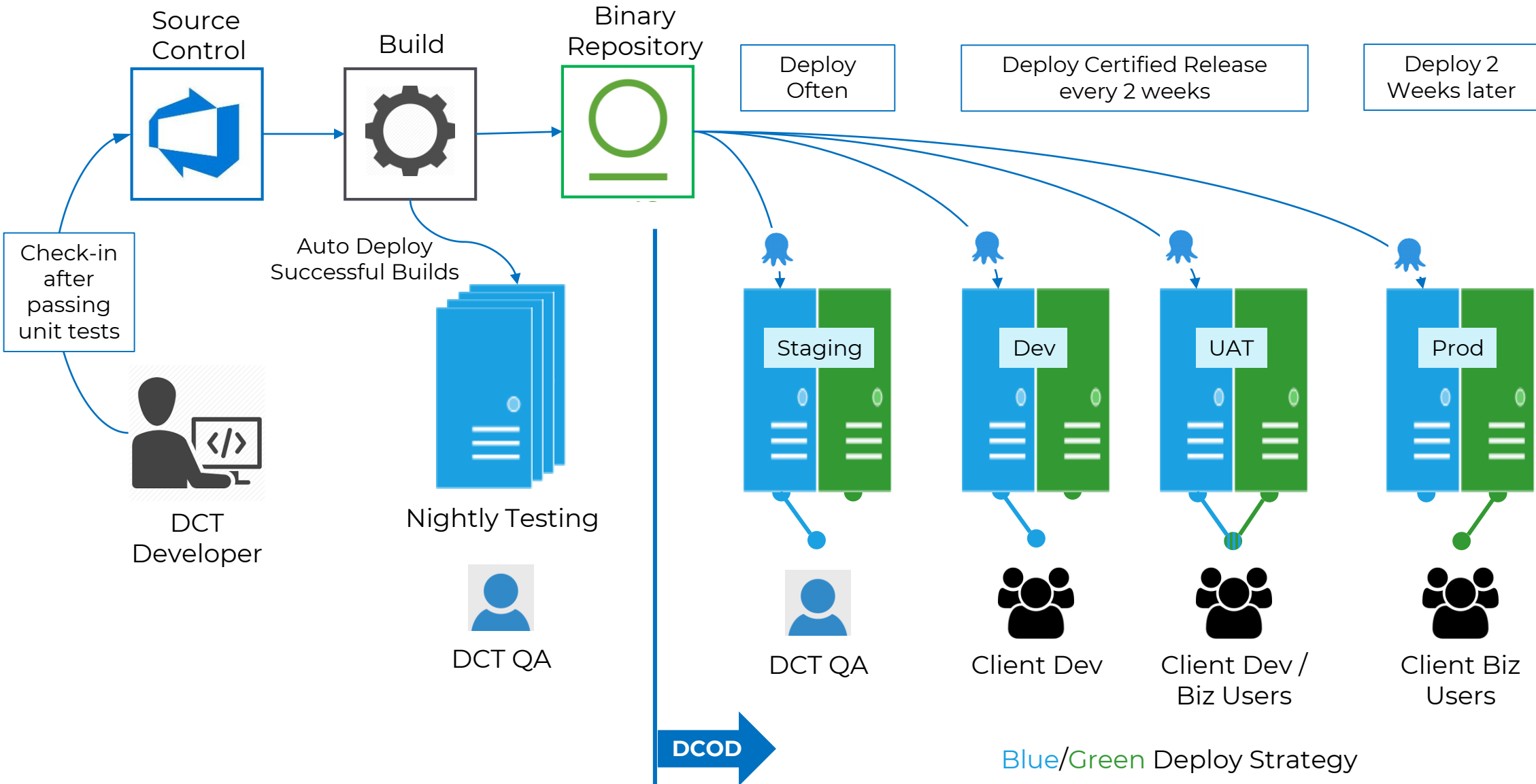
- ▶ Separate automated deployment process for client configurations
- ▶ Separate (or eliminate) deployment for DC code & client extensions
- ▶ Develop good automated testing coverage
- ▶ Run automated testing as part of CI/CD process
- ▶ Establish a rapid-cycle-time change control process

Alternatives?

NONE



# Automated DevOps Pipeline / Test Automation



# Preparing for Continuous Updates

The “continuous update” process is *new*, but carriers can still take steps to be ready for this process:

*Follow best practice guidelines we’ve discussed*



# Duck Creek Architecture

