

**BARK**

# From Code to Clicks Transition to Low-Code Engineering & Its Impact on Security



June 2025  
Michael Contino

# Overview

Whoami

**Who is this guy?**

What

**Bark all about?  
Technology Evolution?**

Why

**Low Code or No Code?**

How

**Adapt our security strategy?**



# Michael Contino

## Principal Security Engineer at Bark

### CAREER TIMELINE



Traveling  
Penetration tester  
for 4yrs

 Crowe



Ran Vulnerability  
Management and  
Devopsec for 5yrs

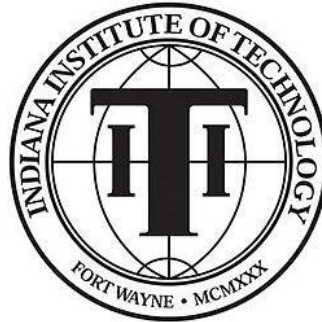
**GROUPON**



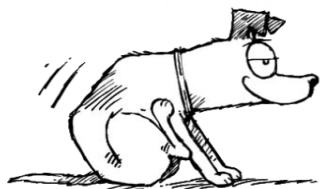
Infosec Tech Lead  
of all the things  
for last 3yrs

**BARK-**

### Alumni of



### Certifications



At BARK, we want to make dogs as happy as they make us. Because dogs and humans are better together.

**BARK**

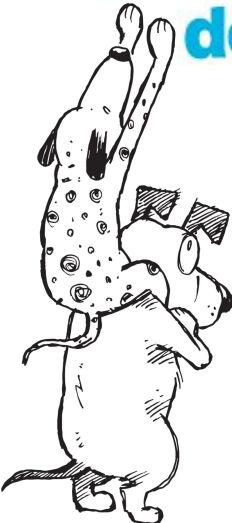
**BARK Air**  
dogs fly first.

**BARK BOX**

**BARK SUPER CHEWER**

**BARK Bright**

**BARK Eats**



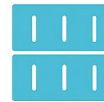
**BARK:NYSE**

# Bark Technology Evolution

## BARK TECHNOLOGY EVOLUTION



Started Monolithic  
Ruby on Rails app  
(BarkBox)



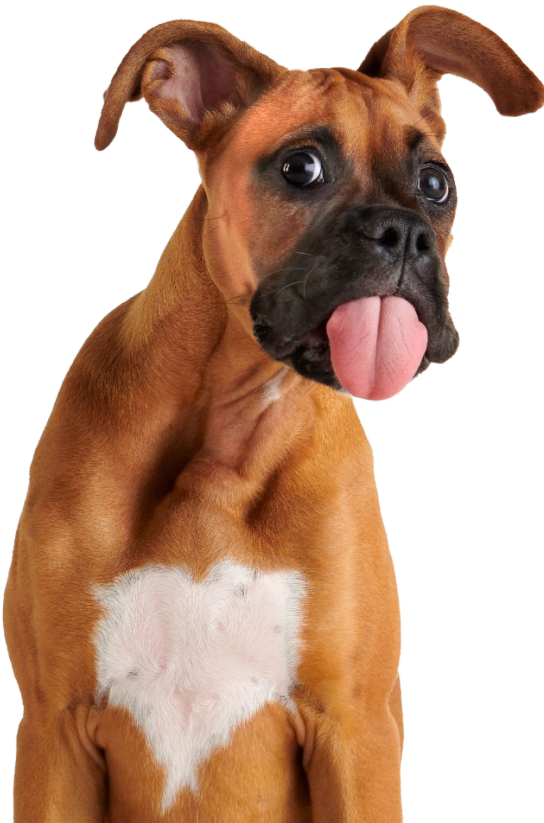
Then Containerization  
of business brand apps



Now Shifted all brands  
to single Shopify  
storefront (Bark.co)



Next We will utilize AI  
empower data-driven  
development



# Benefits of Low-Code/No-Code Platforms



-  Increased development velocity & agility
-  Reduced engineering overhead
-  Faster go-to-market for e-commerce features
-  Lower compliance complexity
-  Shift innovation focus elsewhere



# Security Challenges in Low-Code Environments

## Limited control over infrastructure

Low-code platforms offer limited ability to make custom changes in comparison to operating your own infrastructure. This can make it difficult to apply effective controls outside what is offered directly from the service provider..

## Third-party dependencies & integrations

To compensate for the the lack of customization NCLC platforms often create exensible app/plugin systems to allow others in the community to develop enhancements. This leads to a nested ecosystem of dependencies and integrations that can be difficult to manage.

## Data security and Access control

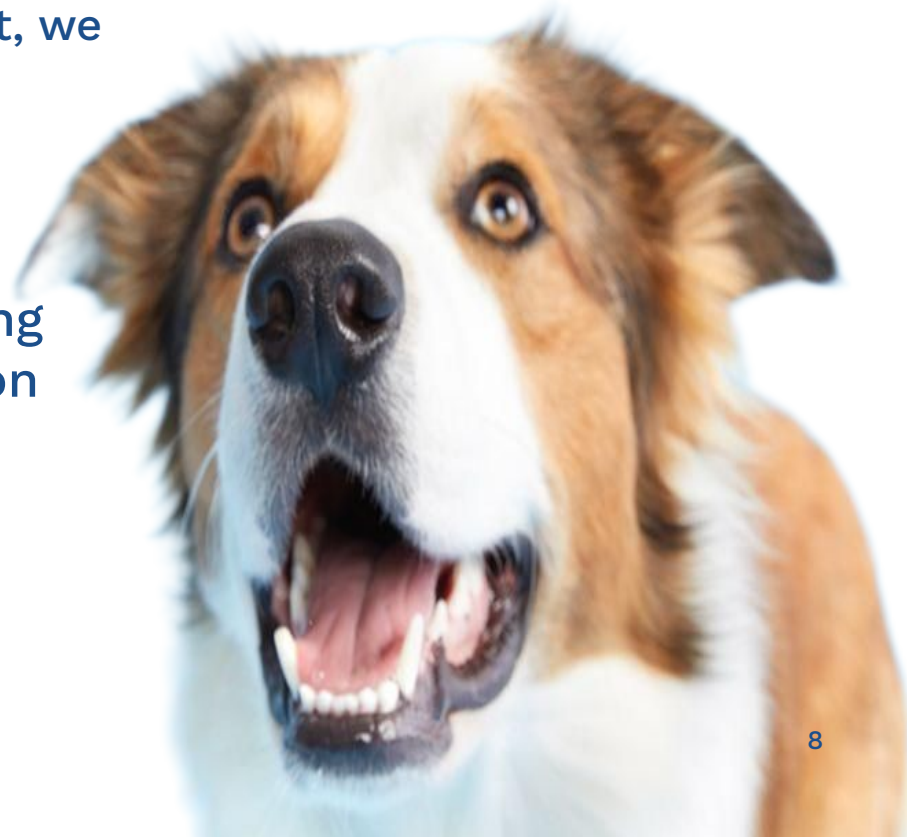
A data rich API is at the heart of NCLC platforms, allowing both ease of access and empowered to conduct business. With just a few clicks you can often enable a new app and grant it access to vast amount of data. Securing that data, controlling access, and understanding where it flows becomes a primary focus.



# How Bark's Security Team Adapted.

In order to adjust our security program to support a transition to a shopify storefront, we have made the following changes.

- Shift focus to Data flow security
- Implemented Key Security Controls
- Established active change monitoring
- Focused Investment in AI automation



# Shift Focus to Data Flow Security.

Challenge full record requests as hard as you would full access.

Protect and monitor sensitive data fields.

Move focus from secure perimeter to a secure workforce edge.

Ensure third-parties understand their share responsibilities to protect data.



# Key Security Controls Implemented.

- Central Identity Provider with Certificate based Device Trust
- Securing Third-party apps with Access gateways
- Web Application Firewall (WAF) & bot mitigation
- Data Access Controls and Classification



# Central IdP and **Device Trust.**



## Utilize central identity provider

**Most IdP provides allow for client based certificate authentication.**

This can be used alongside traditional user authentication and MFA to also independently authentication the connecting device itself.

## Create Your own CA Infrastructure

**Creating or cloud host your own certificate authority is an effective way to establish trust.**

You can then trust the CA with your IdP, and Issue client-auth certificates to each company controlled device.

## Enforce Certificate authentication

**Require certificate authentication from devices before allowing access to sensitive services.**

By requiring cert based auth alongside your user you can better ensure sensitive data can only be accessed by company secured devices.

# Securing Third-party apps with **Access gateways.**

## SAML

**The primary focus should be to authorize everything against the central IdP.** Probably the best and most supported by would be to have your third party utilize SAML integration to establish sessions.

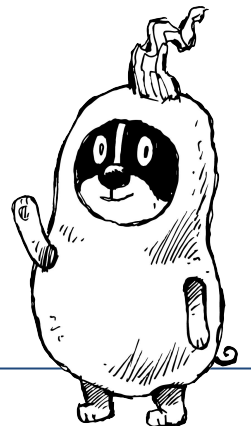
## OAuth

**Alternatively utilize an OAuth2.0 grant against an service that supports SAML.** This can be done to allow integrated services like gsuite or github to act as a hub for authorization, without it turning into a tangled web.

If you can't utilize either, jump in the middle an access gateway as simple as an nginx reverse proxy in front of a service or hosted system you want to protect.

```
server {
    listen    443;
    ssl on;
    server_name myserver.com;
    ...
    ssl_client_certificate /etc/nginx/certificates/Myca-chain.cert.pem; ## Use your own trusted CA certificate from CA/SSLTrust
    ssl_verify_client on; ## Force the proxy to authenticate client with a certificate signed by your CA

    location / {
        ...
        proxy_pass https://service.com
    }
}
```



# Rethink Web Application Firewall and Bot Management

Your low-code No-code (LCNC) providers web protection goals are different than yours!

It's still your companies responsibility to combat fraud and automated account abuse.

Your WAF protections and edge service provider is still the best place to shape and control inbound traffic.

Most edge provider have the ability to utilize behavior analytics to manage bots and automated traffic.

Bot management combined with advanced rate limiting, can be an effective way reduce automated account activity and fraud.



# Data Access Control and Classification.

If you haven't already classify your data points and **whole records**. Key combinations of data points can become more sensitive or damaging to your brand.

Assign a specific **set controls to each classification** of data, not services or datasets. If sensitive data lives or works there it shall be protected.

Document **where sensitive data works** and which **access identities** work with it. Monitor those **devices** and users grants and access requests to limit transport.

Where at all possible, limit the access to sensitive data to read only and **delegate write permissions** to a non-human identity.



# Active **Change Monitoring** in Shopify.



Shopify  
UI



Shopify  
Admin API



Cloud  
Events



Slack  
Message  
(Admin Change)

## Monitor admin clicks as events in the Admin API

**Shopify creates events in the GraphQL Admin API for each change requested in the UI.**

We can create a service to monitor these events in near real time and parse them into a structured front like cloudevents to be supported by automated tooling.

## Alert changes and permission grants

**Target events like app installs or elevated permission grants**

Once an event is seen in cloudevents based on our search for app changes or grants we trigger a structured message to a slack alerts channel to notify the team.

# Future Investment in AI Automation

## Development

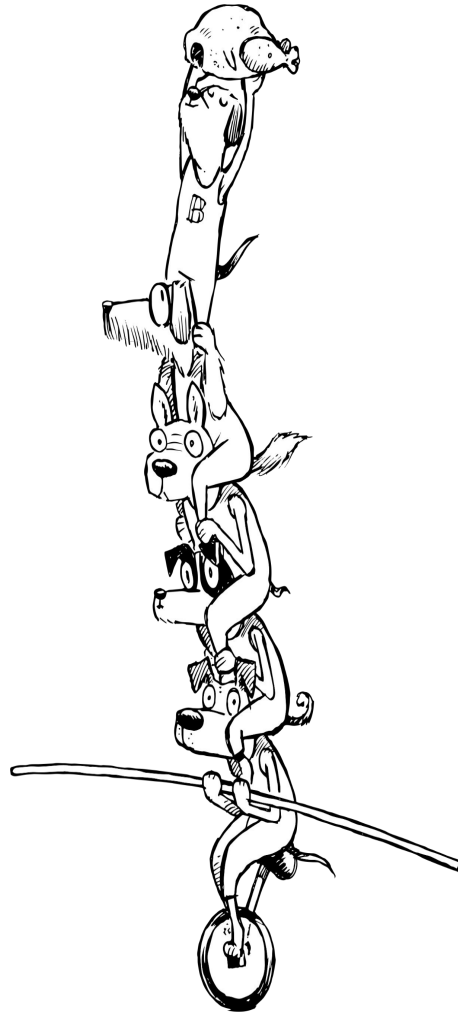
### Increase agility with Github copilot auto-remediation

Even with large development shifts, there is still data to be managed and things worth creating that require protection.

## Communications

### evaluate emails and communications for suspicious with Smartphish

Business happens in a multitude of ways and we operate in a social industry. Normalizing and then screening communications for suspicious requests is key.



## Traffic Shaping

### Identify traffic patterns for fraud and behavioral management

Not only does utilizing a WAF and bot management reduce risk of fraud and abuse. It also keeps analytics cleaner for more reliable marketing.

## Data Protection

### Empower DLP with automatic data discovery

Data is everywhere and the fuel for growth. We do our best to keep to controlled, but detection is key for when things bounce outside the norm.

# Conclusion & Takeaways



## **Low-code adoption requires a shift in security mindset to follow the data**

Enriched data is a key factor that provides the competitive edge in a shared Low-code platform. Which leads to a higher vigilance and control of how data is leveraged.

## **Many existing security technologies can adapted to still meet needs of a low-code platform**

There is no need to reinvent the wheel, just utilize the tools in slightly different ways. Many core technologies like device trust are still very relevant.

## **Focus practical security controls on the edge where users are actually doing work**

Low/No code solutions and there extensible nature reduce the reliance on trusted networks. Secure controls have to be designed for where work happens.

# Thanks!

Linkedin: /in/mcontino

@sleventyeleven

mcontino@live.com

