Microservices: Patterns for Infrastructure Modernization with Express Serverless Platform

Prepared by LunchBadger

Introduction

When planning an approach to modernizing legacy applications, there are alternatives to a full re-write and cut over. Each approach has it's own implications for the business and developer resources.

A rewrite means higher risk and longer time to value. This can tax your development resources and slow down engineers who could spend that time delivering value to customers. Less agile, slower velocity. It's just not always a practical approach.

A more compelling approach is to leverage the “strangler” pattern. Despite it's bizarre name (which we can totally relate to at LunchBadger), this modernization pattern for infrastructure development was introduced by Martin Fowler in 2004. Fowler was interested in how developers can give value steadily through frequent releases where engineers could monitor progress more carefully.

The Strangler Pattern is an application design pattern that transforms your monolithic application into microservices at an incremental level.

This pattern replaces a functionality with a new service and works as follows:

- New functionality is ready.
- Old, out dated component is "strangled."
- New service is put into production.
- Old component is decommissioned altogether.

We've put together this article to discuss important aspects of how Express Serverless Platform empowers developers to easily implement this modernization pattern.

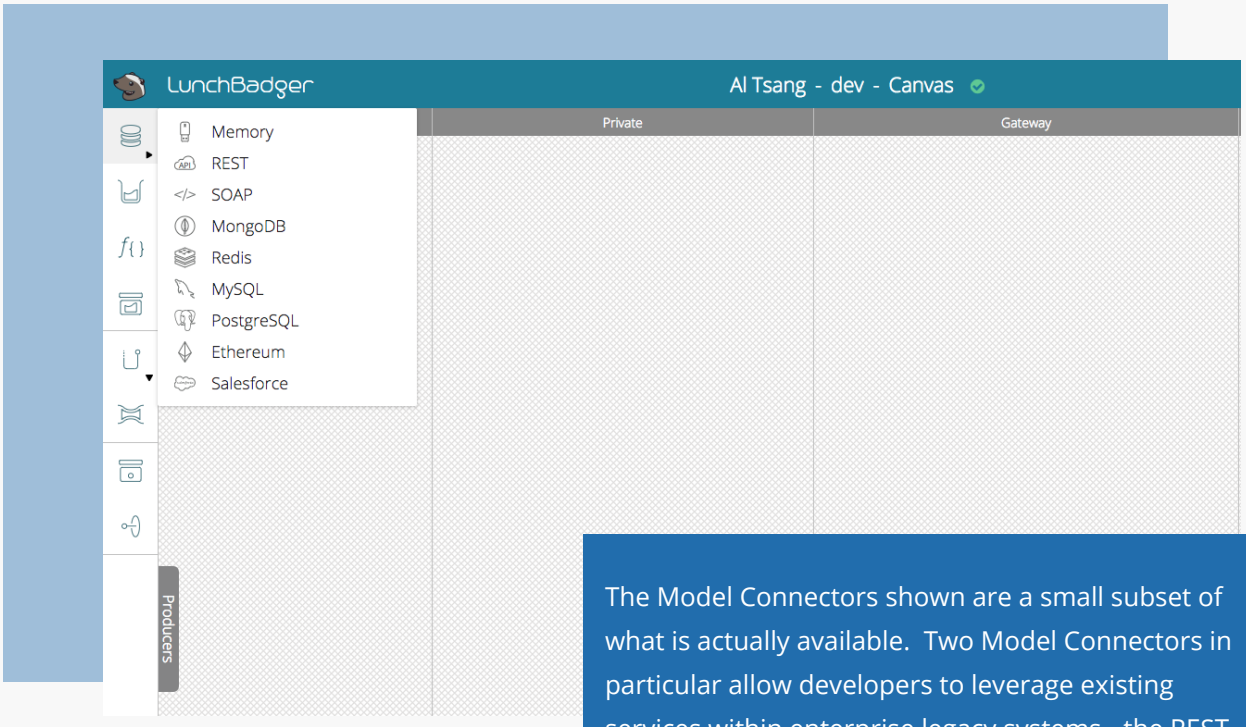
> "Reducing the complexity of an application enables you to deliver business features faster. It also allows you to scale your application based on increasing load. Having an automated CI/CD pipeline makes it a lot easier to deploy the microservices and can make the transition from monolith to microservices much smoother." - Samir Behara

Pattern Implementation Considerations

Express Serverless Platform (ESP) provides cloud tooling that reshapes a company's legacy system into more agile, maintainable and automated discrete parts that run in the cloud. Running in the cloud enables a company to provide its services digitally and securely through modern protocols and interactions that can keep up with rapidly changing needs of customers. Express Serverless Platform not only provides the tooling to allow developers in a company to perform their job faster but also has the best practices on how to do so "built in" so that there is path forward already paved.

A key differentiator of Express Serverless Platform is the ability to start with what you have, build a new set of modern cloud applications and functionality that enables the eventual retirement of such systems piece by piece. Express Serverless Platform runs in any private or public cloud. It's runtime takes full advantage of infrastructure resources while masking their complexity from the user.

Previewed below: Express Serverless Platform comes with a vast set of Connectors for microservice level integration to do this.



The Model Connectors shown are a small subset of what is actually available. Two Model Connectors in particular allow developers to leverage existing services within enterprise legacy systems - the REST and SOAP Connectors.

Introduce a new clean microservice layer

Express Serverless Platform has special functions known as Models. Models are written solely in Node.js/JavaScript. Node.js is an extremely powerful technology that is particularly well suited for not only writing function based microservices, but also doing asynchronous and synchronous integration to multiple systems.

A Model is a function that "models" a real life object and has built in behavior that is commonly required in most enterprise application use cases.

Expose your microservice as an API

With a clean microservice layer built with three model functions, developers can now externalize the microservice as an API that can be consumed by any digital channel that they plan to utilize.

To expose a microservice as an API, the API must be secured by a critical piece of infrastructure known as the API gateway. The API gateway controls access to the API through security policies and also can control the consumption of the API through other policies known as quality of services such as rate limits and quotas. Express Serverless Platform utilizes Express Gateway as its built in API gateway.

Express Serverless Platform is the first of kind platform that provides a seamless orchestration of application and infrastructure microservices all in the same runtime. This allows the bank's operations to standardize on a common runtime natively built for the cloud based on container technology.

In Express Serverless Platform, deploying and configuring an API gateway is a click of a button and utilizes the same Canvas to illustrate its relation to the microservice. Once a gateway is deployed, a microservice can be exposed as an API and a set of endpoints by simply dragging a line from the model to the gateway's pipeline

Secure and manage your APIs through pipelines and policies

Express Gateway is easily configured through pipelines. A pipeline is a set of policies that can be customized to control access to the API. A single Express Gateway instance can be configured with multiple pipelines. Developers can quickly configure multiple policies including key authorization and header manipulation policies to shape the API request to be universally consumed by the application microservice that it is proxied to.

Any number of API endpoints can be declared and externalized giving the flexibility of creating different ways of consuming functionality by different clients. A set of endpoints can be associated with different pipelines to meet differing needs of security of service.

APIs are the “omnichannel” backend interface that can utilized by all clients including:

- web applications
- mobile clients
- system to system integrations
- devices

Digitizing and extending their service capability as a digital asset enables them to extend their market reach and open an additional revenue channels within specialized markets where partners already have a captive audience and customer base.

On-demand automation through serverless functions

Express Serverless Platform has two different types of functions - model and serverless functions. Serverless functions are barebone functions that can be written in any commonly used languages including: Node.js, JavaScript,Go,Python,PHP,Ruby,.NET, Core, and Java.

Through Express Serverless Platform, developers can utilize its built in serverless engine for its on-premise deployment and any public cloud’s proprietary serverless offering for a true hybrid and multi-cloud solution.

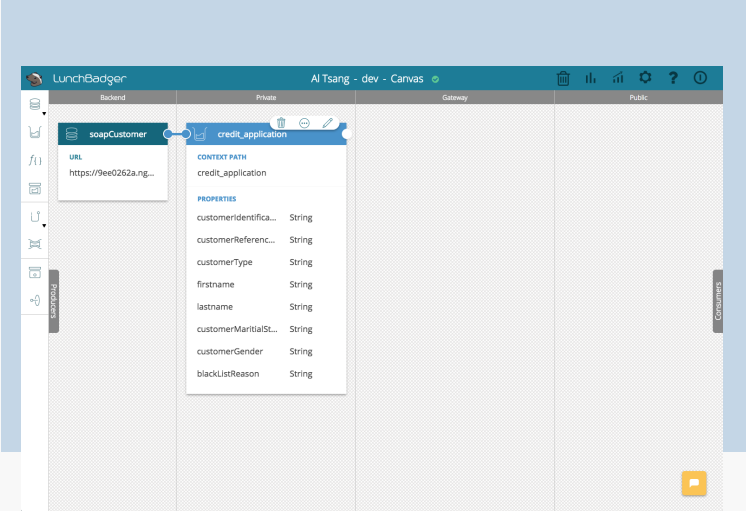
Summary

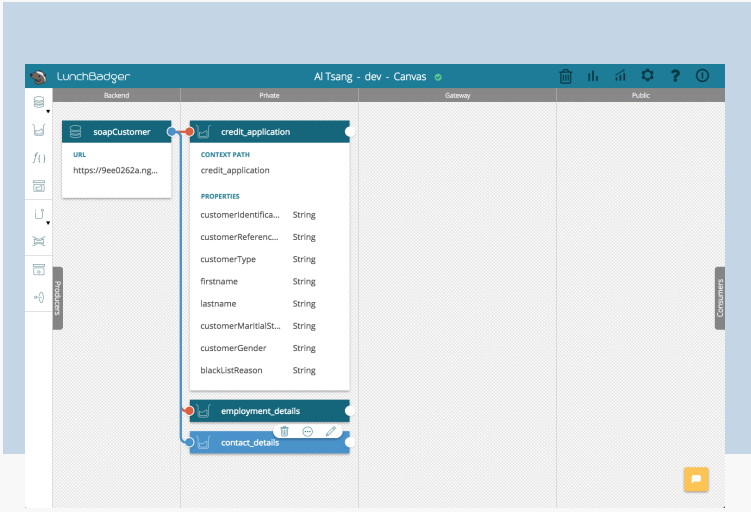
Using Express Serverless Platform, the developers can implement the modernization patterns like Fowler's "strangler" pattern easily and quickly bringing to market a new microservices, functions and APIs leveraging their existing applications. Application microservices can be iterated on quickly to accommodate changing business requirements related to the use case without having to make core changes to the legacy code that would require a cumbersome revisioning of their system.

Furthermore, gaps in functionality that are not supported by any legacy system can be written in more lightweight microservices entirely on the Express Serverless Platform.

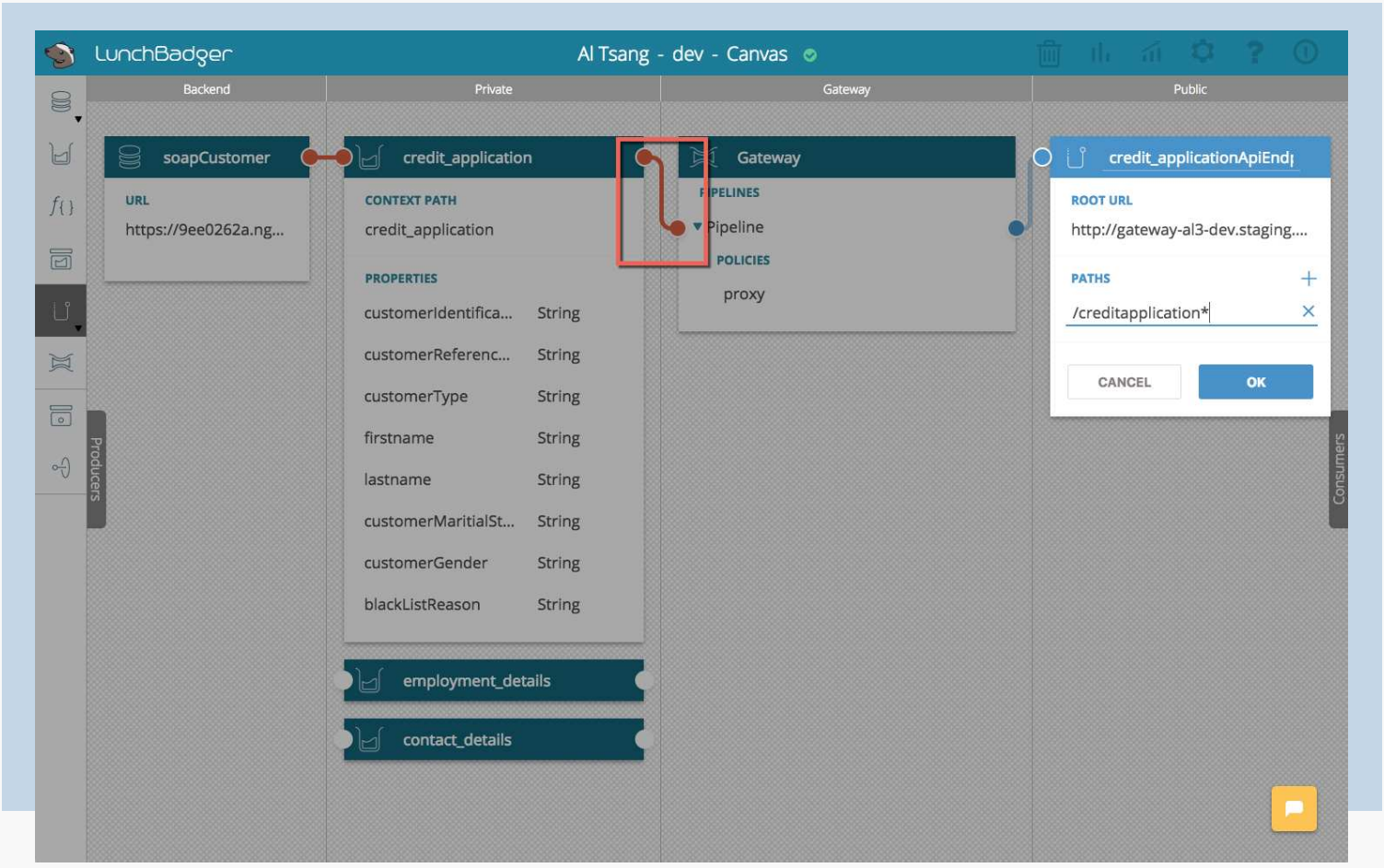
Express Serverless Platform provides a huge time savings and reduction in cost for labor by automating legacy business processes that were previously hard to displace. Express Serverless Platform provides function based technology to utilize other cloud services.

Introduce a new clean microservice layer

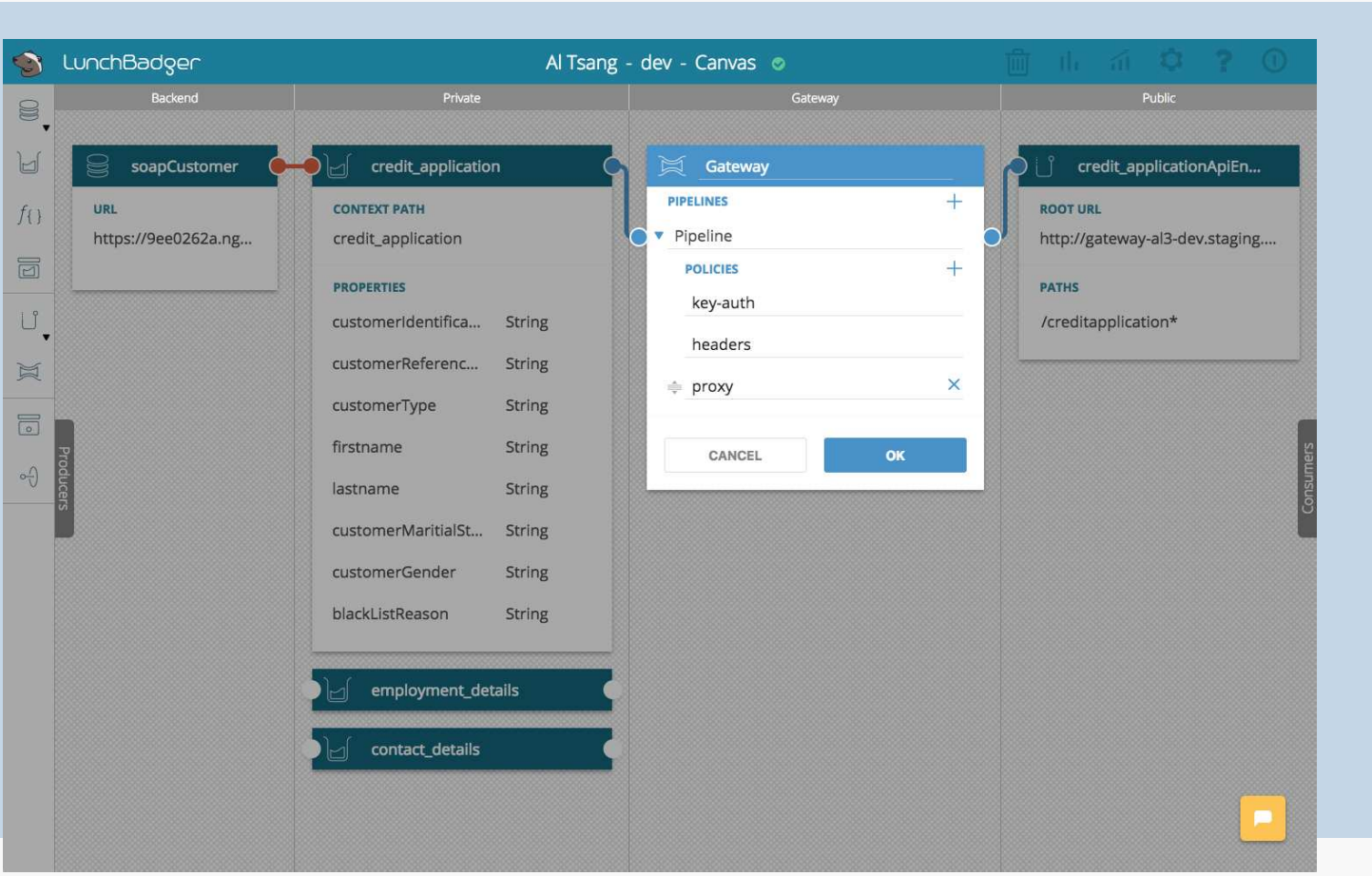




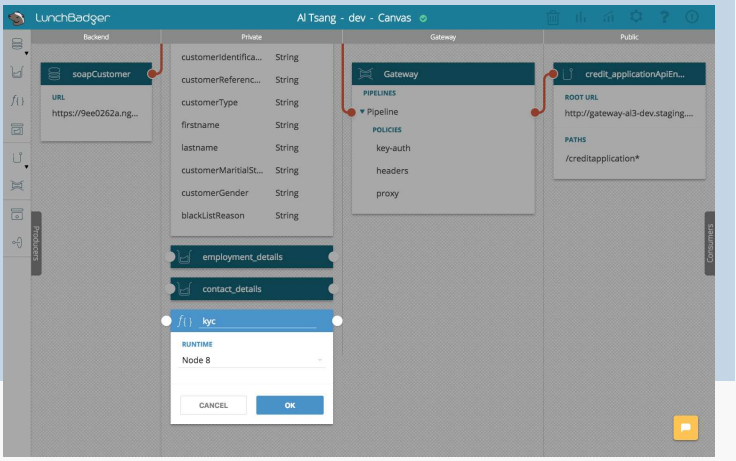
Expose your microservice as an API

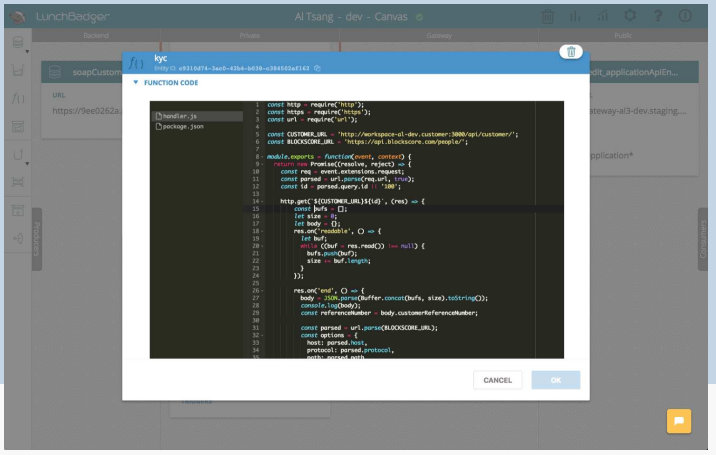


Secure and manage your APIs through pipelines and policies



On-demand automation through serverless functions



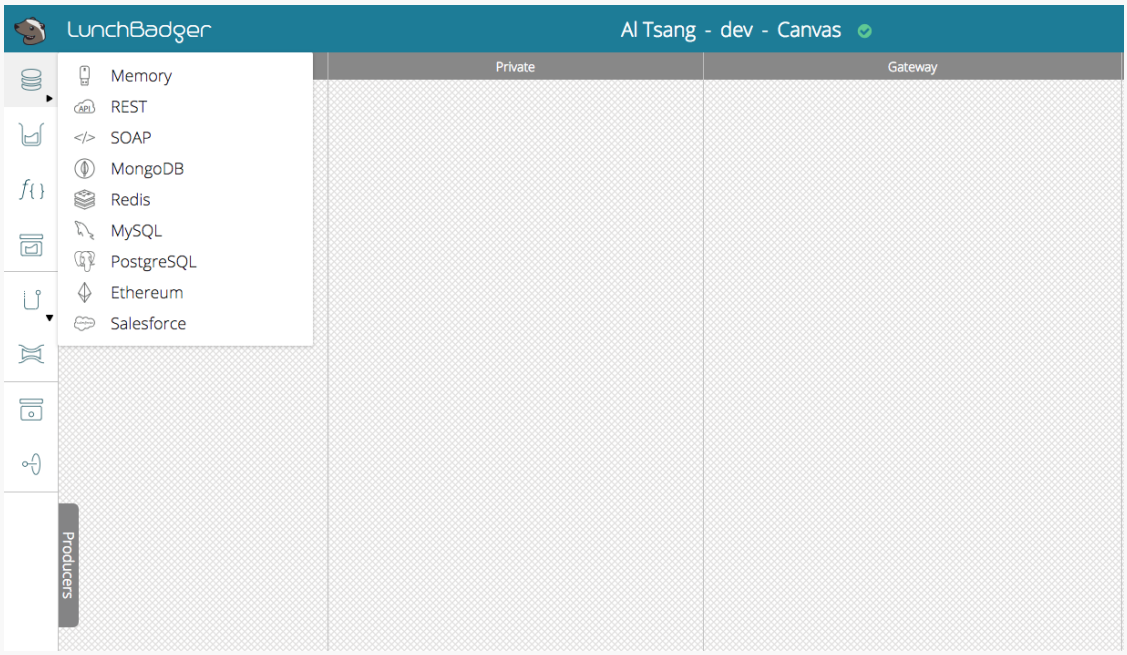


Case Study: Bank Credit Application

Express Serverless Platform provides cloud tooling that reshapes a company's legacy system into more agile, maintainable and automated discrete parts that run in the cloud.

A key differentiator of ESP is the ability to start with what you have, build a new set of modern cloud applications and functionality that enables the eventual retirement of such systems piece by piece. Express Serverless Platform runs in any private or public cloud. It's runtime takes full advantage of infrastructure resources while masking their complexity from the user. ESP comes with a vast set of Connectors for microservice level integration to do this.

The bank's credit application utilizes the Customer SOAP service. This SOAP service represents all customers in the bank and contains more than 160 fields of information. With a click of a button Express Serveless Platform can easily connect to such SOAP service and provide a set of tools to interface with it.



When it comes to modernization, some of the most prominent concerns are cost and application design. How can a business reduce costs by becoming more efficient? Additionally, what are the implications on core application design and microservices? Let's talk!

Concerns on Modernization Cost

> "Many people still don't consider a strangler since they think it will cost more - I'm not convinced about that. Since you can use shorter release cycles with a strangler you can avoid a lot of the unnecessary features that cut over rewrites often generate." - Martin Fowler

In Fowler's detailed explanation, he reasons that because you can use shorter release cycles with a strangler pattern, you can avoid unnecessary features. These features are just the result of a cut over rewrite. Shortening the release cycle also helps to keep your agility and velocity while developing applications.

Application Design & Microservices

> "Reducing the complexity of an application enables you to deliver business features faster. It also allows you to scale your application based on increasing load. Having an automated CI/CD pipeline makes it a lot easier to deploy the microservices and can make the transition from monolith to microservices much smoother." - Samir Behara

Applying the strangler pattern involves writing new pieces of functionality surrounding the monolithic application. Using a gateway, calls can be routed to the new pieces rather than the legacy code, over time, strangling the legacy with newly developed pieces. Microservices provide an ideal development paradigm for implementing the strangler pattern. Microservices help with breaking down application to micro, autonomous parts as well as promoting the separation of concerns principle to encapsulate functions of the application. Combined with an automated CI/CD pipeline, a microservices approach provides a high velocity capability and value delivery environment versus a monolithic application rewrite approach.

How does LunchBadger's Express Serverless Platform (ESP) deliver these capabilities and support the use of modernization patterns like Martin Fowler's "strangler" pattern? Stay tuned for an active case study or, you can try Express Serverless Platform free for 14 days.