

Service Architecture

As you build larger and larger systems, individual components can be developed within the same system or separated out.

[Monolithic Architecture]:

[Microservice Architecture]:

[Serverless Architecture]:

Monolithic	Microservice	Serverless

Example: CoinFlip

In our game of CoinFlip, players begin with 100 IlliniCoins and spend 1 IlliniCoin each time to play CoinFlip:

- If the coin lands on “Lincoln” (heads), they get 2 IlliniCoins.
- If the coin lands on tails, they get 0 IlliniCoins.

Q: What are the different components of this system?

Configuration and Deployment Challenges

One of the most challenging bits of microservices is managing the configuration and deployment of the microservices:

- What is the location of my dependencies?
- How do I quickly update the configuration?

Solution: _____

Every process on every Operating System runs with a number of **environmental variables**.

-

Command to List All Environment Variables	
Linux:	env
Windows PowerShell:	dir env:

Common Environment Variables

Common Programming Convention: .env Files

A common, but not built-in, programming convention is to use .env files to specify deployment-specific environment variables.

.env file
FLASK_RUN_PORT = 24000

...now, when we run Flask, we see it starts on a different port:

\$ python3 -m flask run
[...]
* Running on http://127.0.0.1:24000/ (Press CTRL+C to quit)

Building CoinFlip

Database:

\$	docker run --rm -it -p 27017:27017 mongo
----	--

CoinFlipGameService:

- Route:

Functionality:

IlliniCoinService:

- Route:

Functionality:

- Route:

Functionality:

CoinFlip:

- Route:

Functionality:

- Route:

Functionality: