

The transition from one main chain to many shards is similar to the shift from monoliths to microservice architecture. Due to lack of consistency, code written in a microservice environment has to be written fundamentally differently. Inability to perform atomic transactions or joins across disparate data sources introduces difficulty for developers. To compensate for this increased difficulty, modern programming languages have introduced new primitives to operate in an asynchronous world.

As ethereum reaches new levels of scale, a similar shift for developers to fundamentally change how they write contracts will need to happen. This will require new abstractions to make the developer experience more ergonomic. I propose that javascript's "async/await" syntax could be built added to solidity for this purpose.

For example, imagine that USDX is deployed in shard X and USDY is deployed in shard Y. A contract deployed in shard X could offer trades based upon the value of USDY in shard Y.

```
async function exchange(uint amount) { uint currentValue = await YContract.getUSDYValue() // transfer USDX for Eth based on value }
```

In this example, the contract execution gets suspended while a message is sent and the reply received from shard Y. When the reply is received, the execution picks up where it left off. Similar to how async/await works in javascript, the caller contract must also be async.

This is a simple example, but I think this is an important topic to begin thinking about in more detail. Right now, sharding is highly abstract and many people I talk to are under the impression that it will magically allow the ethereum main chain to scale without any changes in how contracts are written.

In addition, fleshing out the more practical details about the developer experience can help better make core design decisions. For instance, we might think differently about [the idea that shards will be evenly distributed](#). Different contracts will be deployed in different shards. Contracts will depend on each other so ecosystems within each shard will form with different fee markets. Organically as fees increase, users will migrate to other shards.