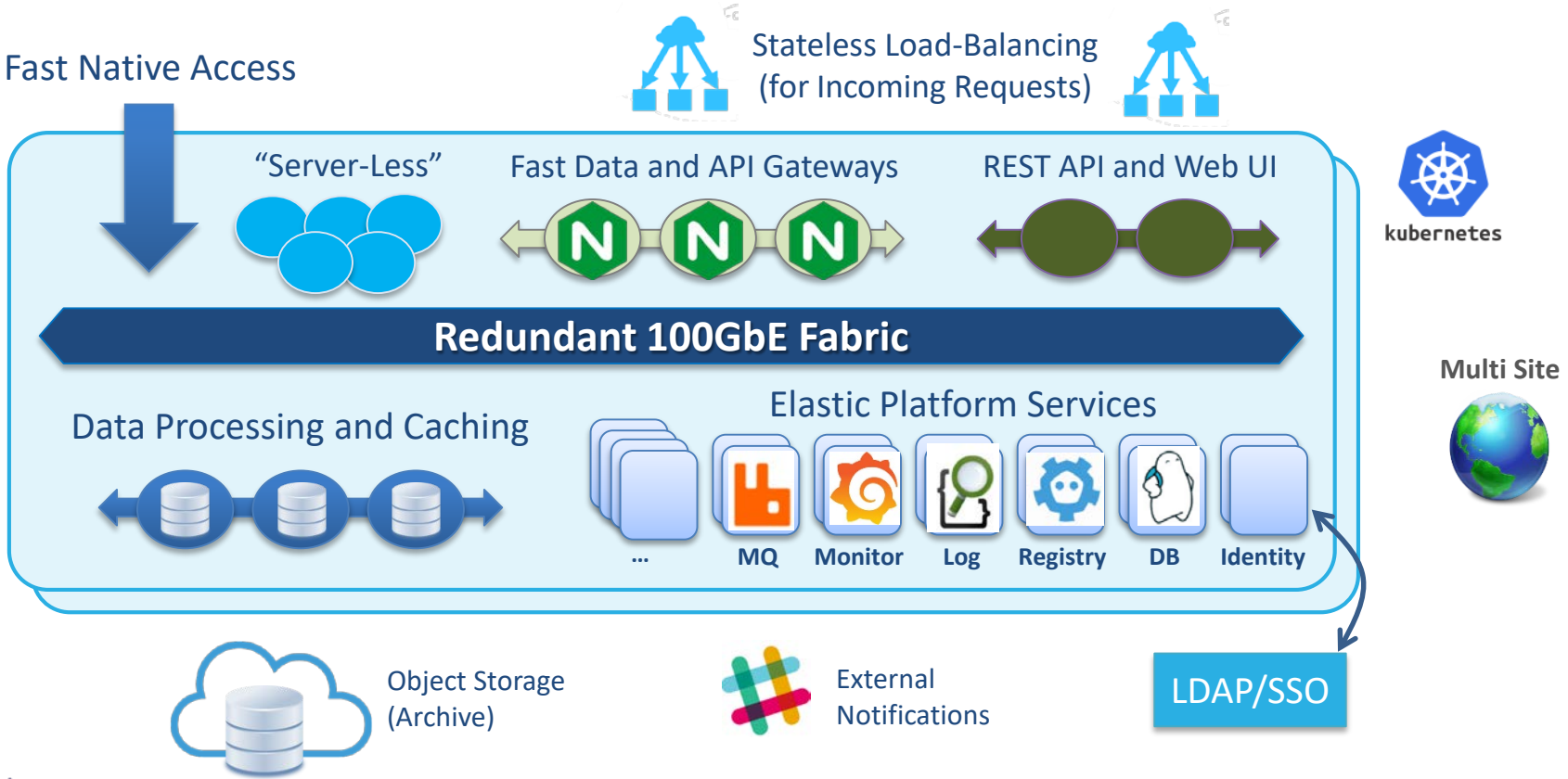
Building Super Fast Cloud-Native Platforms

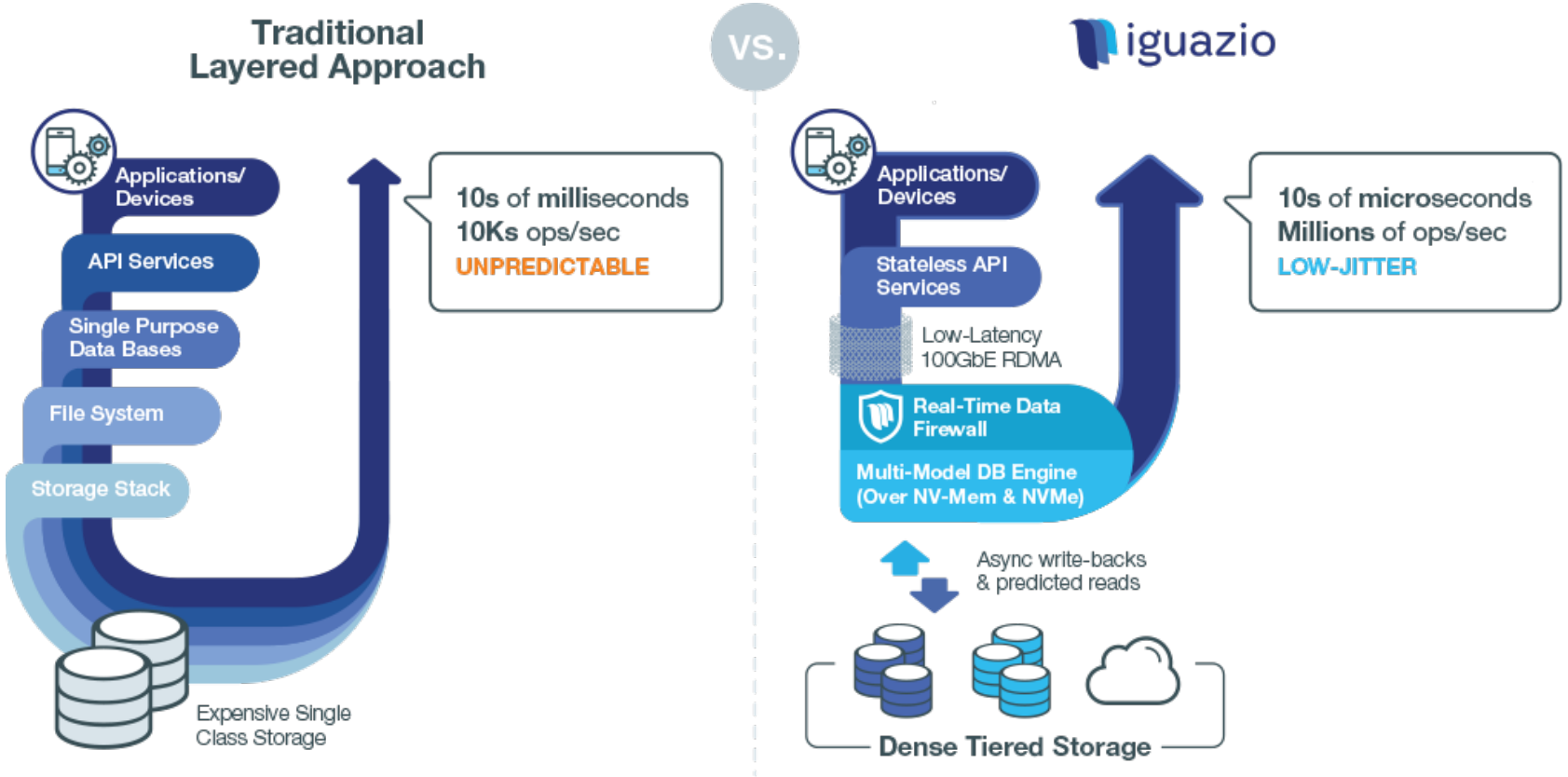
Yaron Haviv, CTO, iguazio
@yaronhaviv

March 2017

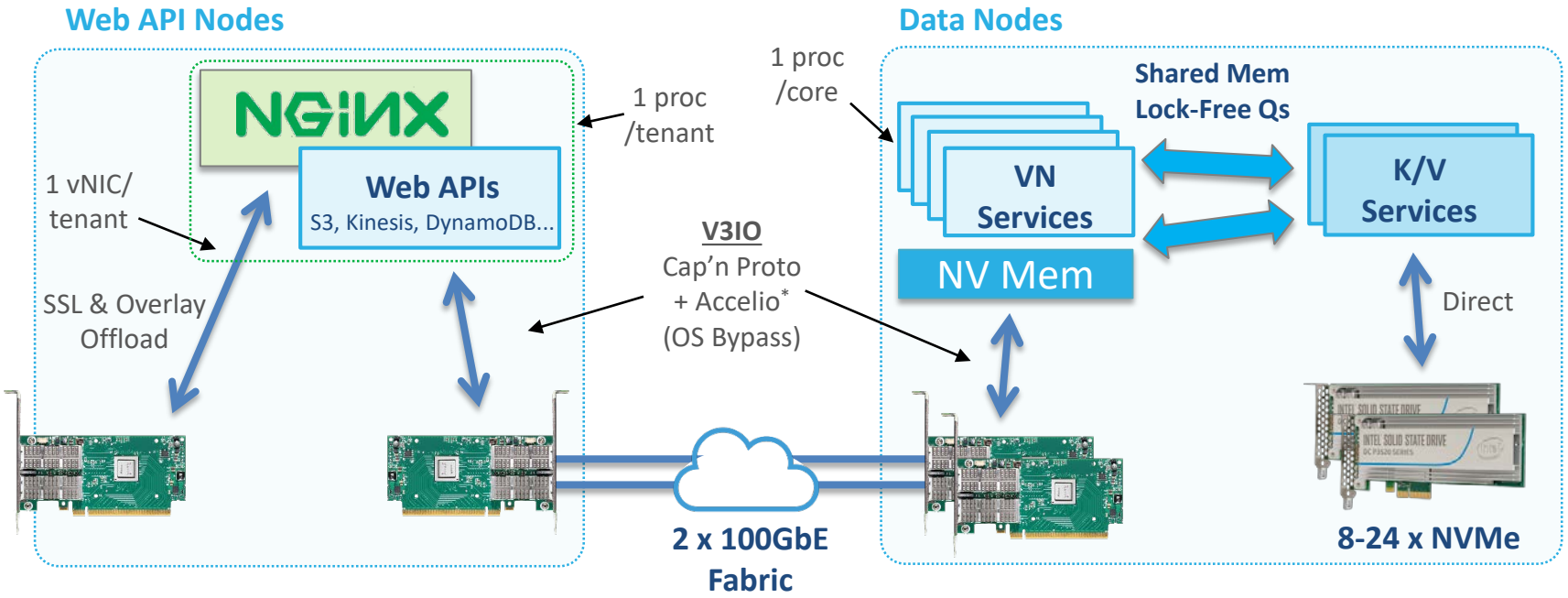
Building A High-Performance Cloud-Native Data Platform



Redefining the Stack, Delivering Magnitudes-Faster Performance



High-Performance Requires Careful Hardware Integration



Serve 800K** Web Req/Sec

Serve 2M Ops/Sec and 100Gbps
0.1ms Latency @ 99% (percentile)

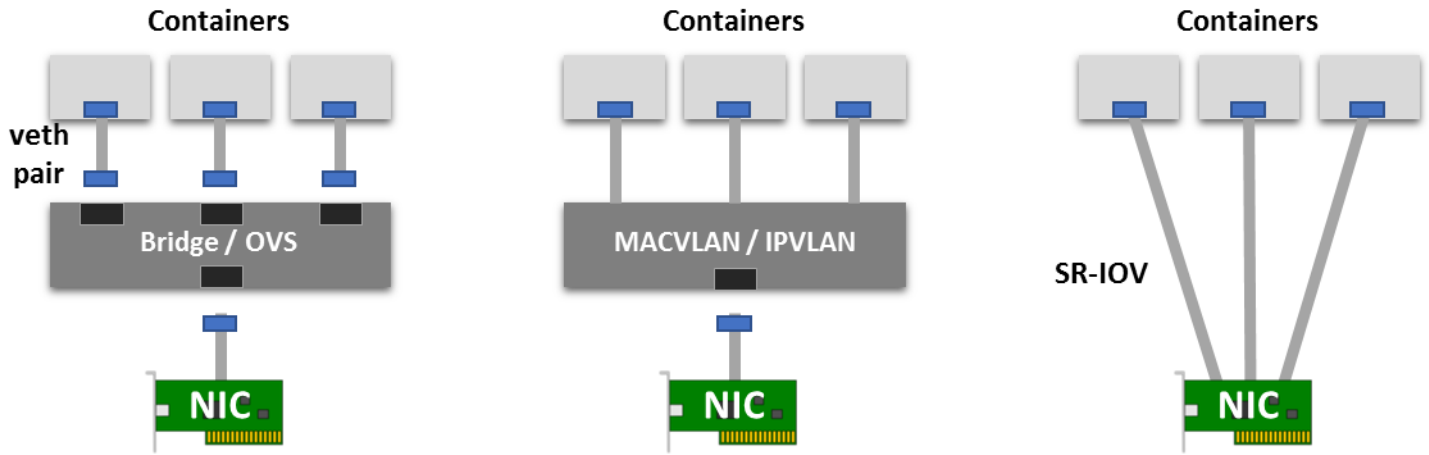
* Accelio: https://github.com/v3io/accelio

** Tested with: https://github.com/v3io/http_blaster

Challenges with Containers and Kubernetes

- Kubernetes Limitations/ Challenges
  - Only one IF per POD, usually going through a slow overlay layer
  - No native support for HW NICs (SR-IOV)
  - Hard to expose low level drivers/ libraries to container
  - Hard to use shared memory IPC/ files between PODs
  - Docker and Kubernetes are different (security, net, volume, shmem, ..)
- Solution
  - Custom network (CNI) and volume drivers
  - Use granular privileges
  - Many trials and errors ☹

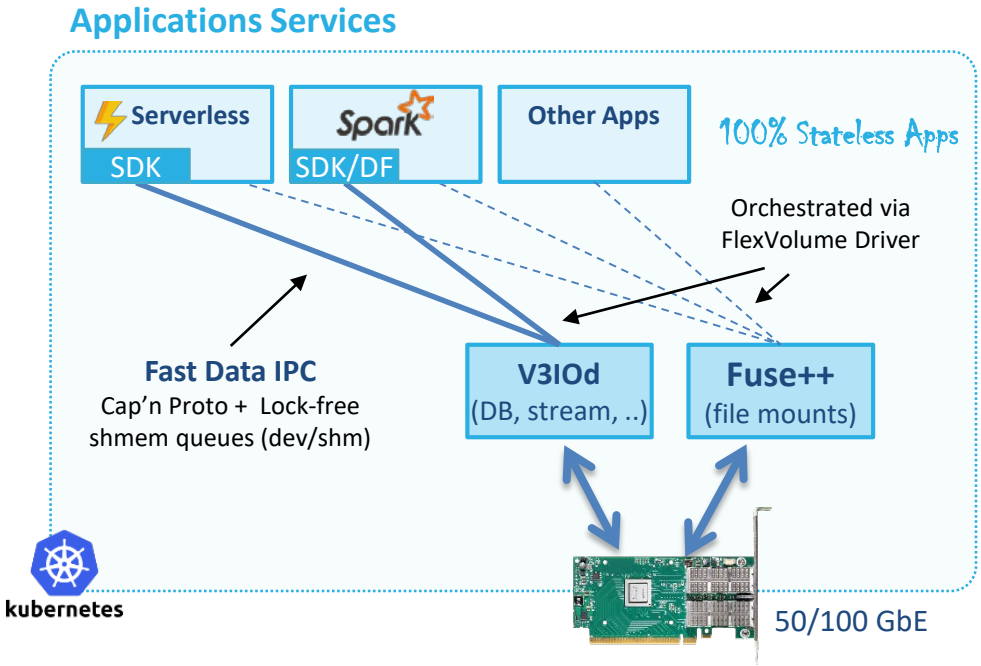
Container Networking 101



Container Networking Options

- SR-IOV allows native hardware access
- Multus enable multiple IFs per POD: https://github.com/Intel-Corp/multus-cni
- More details on my blog https://thenewstack.io/hackers-guide-kubernetes-networking/

Stateless Apps with Fastest Unified Data Access



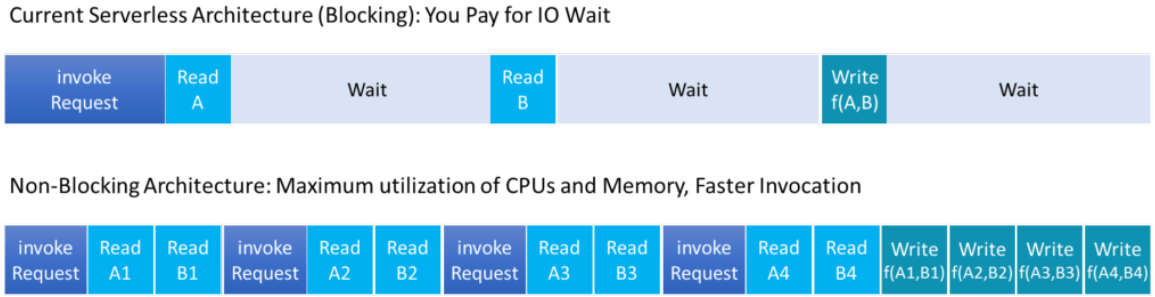
Fuse++ - Modified fuse lib to run async, 15x faster (~100K IOPs/thread)
See: https://github.com/v3io/libfuse

Accelerate Performance Using Shared Memory

- Fastest messaging/DB/obj access
  - Like Go channels across processes
  - Lock-free, async, parallel
  - Zero-copy end to end
  - + Native Spark DataFrame API
- Fast container initialization
  - No TCP/IP connections init
  - No memory alloc/register
- Share TCP/RDMA connections
- NO Kernel drivers/changes !

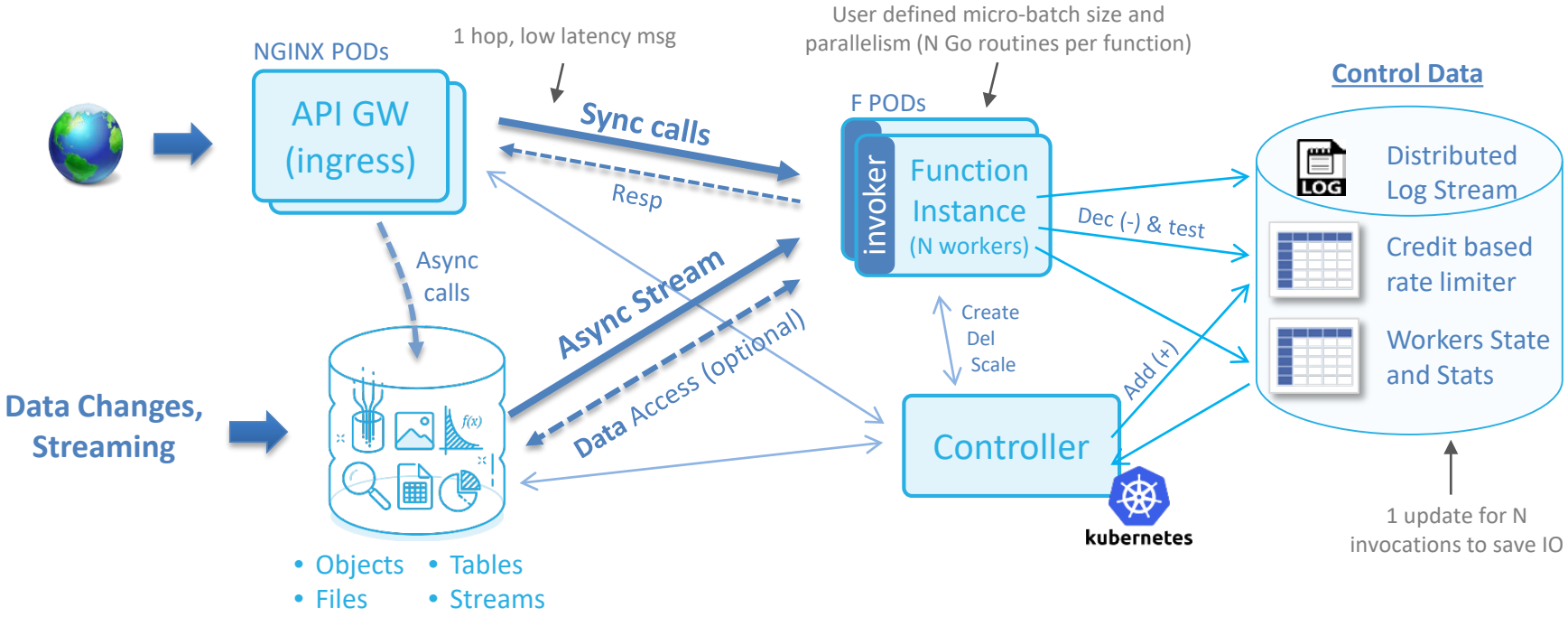
Today: Server-Less is Cool, But Inefficient

- 1st generation is slow to init and a resource drain
  - https://medium.com/@ferdingler/aws-lambda-no-thank-you-9c586990e67d
- Complex and unsecured data bindings (performed in the init part of the function)
- TCP/IP or DB connections may need to re-establish on every invocation
- Slow, limited concurrency, runs one task at a time per container
- Events structure has no common schema
  (see: http://docs.aws.amazon.com/lambda/latest/dg/eventsources.html)



Source, my blog: https://medium.com/@yaronhaviv/serverless-background-challenges-and-future-d0928df71758

Building Server-Less on Steroids



Cut overhead, add parallelism and concurrency, without violating isolation

* Iguazio's server-less framework will be open sourced later this year

Example: Simple HTTP Function

```
func HandleHTTP(event interface{}, wc *Context) (interface{}, error) {

    req := event.(Request)
    wc.Log.Debug("Got Request: %s",req.URL.Path)

    // Read a response text from the bound data source
    body, _ := wc.Data.Get("/path/to/object.html")

    // return some foo response
    return &EventResponse{
        StatusCode: http.StatusOK,
        Body: body,
        Headers: map[string]string{
            "Content-Type": "application/text",
        },
    }, nil
}
```

Built-in log stream

Data binding and credentials in function context
(simple, fast, reusable and secure)

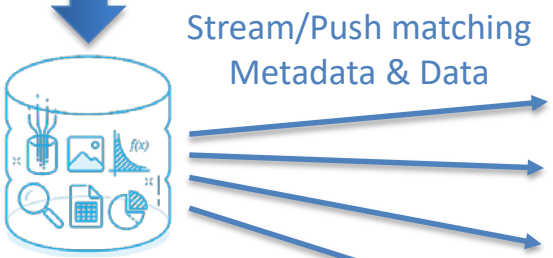
See Data class APIs in backup slide, allow integrating with various sources

High-Speed "Server-Less" Data Processing, Everything is a Stream

Scheduled or **Continuous** Metadata Queries

Select name, mtime, f1, f2, …
Where condition1, 2, 3, ..

Example Tasks:
- incremental **backup** modifies objects/records to **Amazon S3**
- **Scan/convert** content



Control Data
Task stats,
state, and logs

Transactional (exactly once)
update of object metadata
e.g. object last backup/scan time

- Data/Metadata stream is sharded consistently to N **stateless** workers
- Using **StatefulSets** to maintain consistent shard index id (derived from POD name)

kubernetes

Example: Scanning for Sensitive Text Files on Upload/Update

```
var rx = map[string]*regexp.Regexp{
    "ssn"   : regexp.MustCompile(`\b\d{3}-\d{2}-\d{4}\b`),
    "ccard" : regexp.MustCompile(`\b(?:\d[ -]*?){13,16}\b`)}

func HandleText(event interface{}, wc *Context) (interface{}, error) {
    obj := event.([]Record)[0]
    wc.Log.Debug("Processing object: %s",obj.fullpath)

    // Read the file data
    data, _ := Obj.Data()

    // Update Object Attrs + Job Counters based on RegEx
    for k,v := range rx {
        if v.MatchString(string(data)) {
            obj.SetAttrs["has_"+k] = true
            wc.Count["with "+k] += 1
        } else {
            obj.SetAttrs["has_"+k] = false
        }
    }
    return nil, nil
}
```

Init part, e.g. define RegEx filters

1-N records per call

Data/metadata PUSH to reduce latency

Function can use distributed task counters

Async and micro-batch updates of object/record/file attributes

Update data/attrs in one ATOMIC transaction

Image Example, Leveraging a Unified Data Model

```
func HandleImg(event interface{}, wc *Context) (interface{}, error) {
    obj := event.([]Record)[0]

    // Read JPEG EXIF Metadata
    data, _ := exif.Read(wc.Mount + obj.fullpath)
    obj.SetAttrs["Latitude"] = Str2Loc(data.Tags["Latitude"])
    obj.SetAttrs["Longitude"] = Str2Loc(data.Tags["Longitude"])

    // Open file by imaging lib
    img, _ := imaging.Open(wc.Mount + obj.fullpath)

    // Create a Thumbnail and save as attribute
    thumb := imaging.Thumbnail(img, 100, 100, imaging.CatmullRom)
    buf := new(bytes.Buffer)
    _ = imaging.Encode(buf,thumb,imaging.JPEG)
    SetAttrs["__thumbnail"] = buf.Bytes()

    return nil, nil
}
```

Access the objects via file semantics, mount and share handled automatically

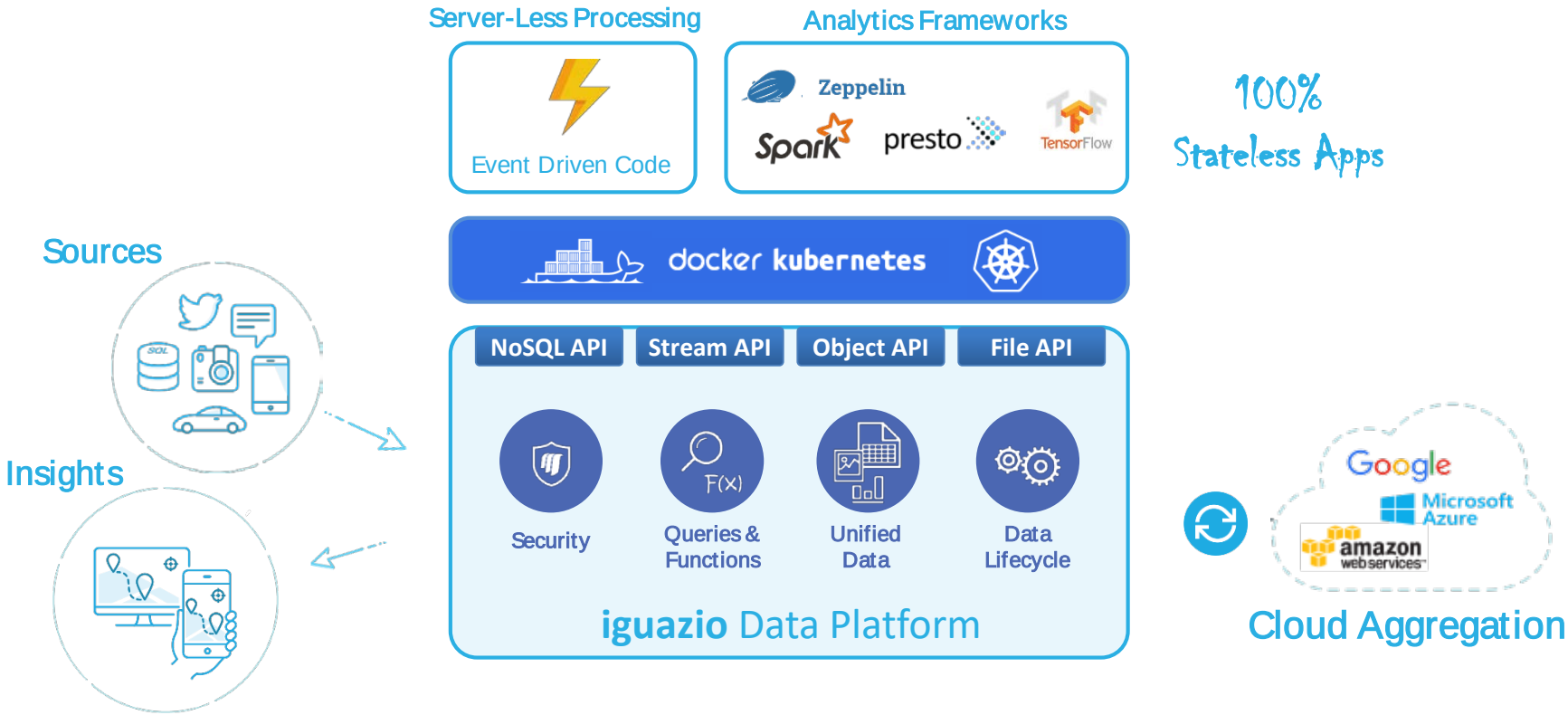
Use standard file access

Thumbnail stored as a small blub record on the **same object** for quick UI access

iguazio

Connecting The Dots: Continuous Analytics Example

Kubernetes Helps us Simplify & Accelerate Analytics at the Edge



Backup

Generic Data Services Binding API

Service	Major APIs
Object e.g. S3, Minio, v3io	**ListBucket**(prefix string) (ListBucketResp, error) **Get**(path string, ranges ...Range) ([]byte, error) **Put**(path string, body []byte ...) ([]byte, error) **Del**(path string) (error)
NoSQL e.g. DynamoDB, Cassandra, v3io	**GetItem**(path, attrs string) (GetItemResp, error) **GetItems**(path, attrs, filter, marker string, ...) (GetItemsResp, error) **PutItem**(path string, list map[string]interface{}, condition string) ([]byte, error) **UpdateItem**(path string, updatestr string, condition string) ([]byte, error) **DelItem**(path string, condition string) (error)
Stream e.g. Kinesis, Kafka, v3io	**GetRecords**(path, offset string, maxrec int) (GetRecordsResp, error) **PutRecords**(path string, records []StreamRecord) ([]byte, error) **Seek**(path string, seek string) (string, error)
File	**Open**(name string, flag int, perm FileMode) (*File, error)