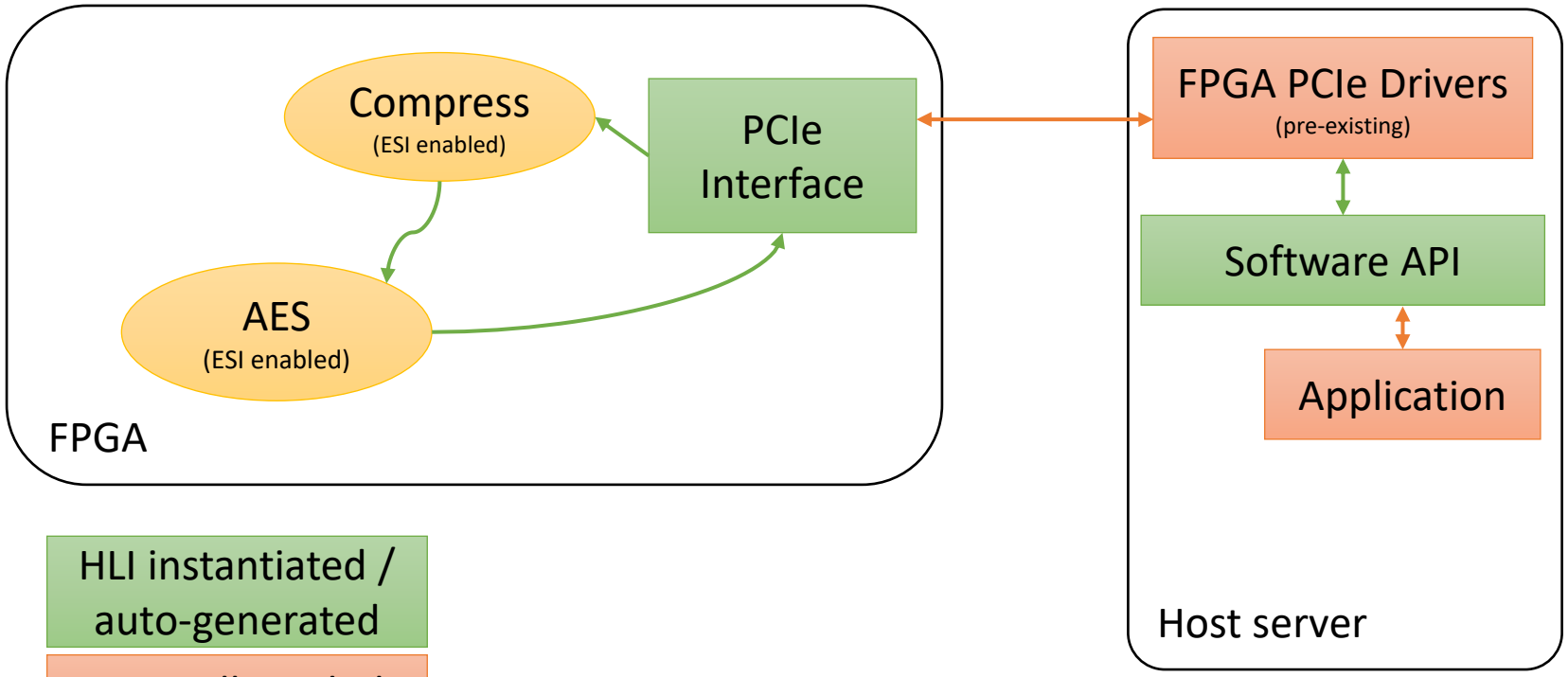
Elastic Silicon Interconnect

John Demme, Microsoft (John.Demme@microsoft.com)

Elastic Silicon Interconnect (ESI) project

- Interconnect compiler designed for:
  - On-chip communication
  - Inter-chip communication
  - Host communication
- Strongly typed
- Latency insensitive / elastic / QDI / FIFO semantics / [your term] …

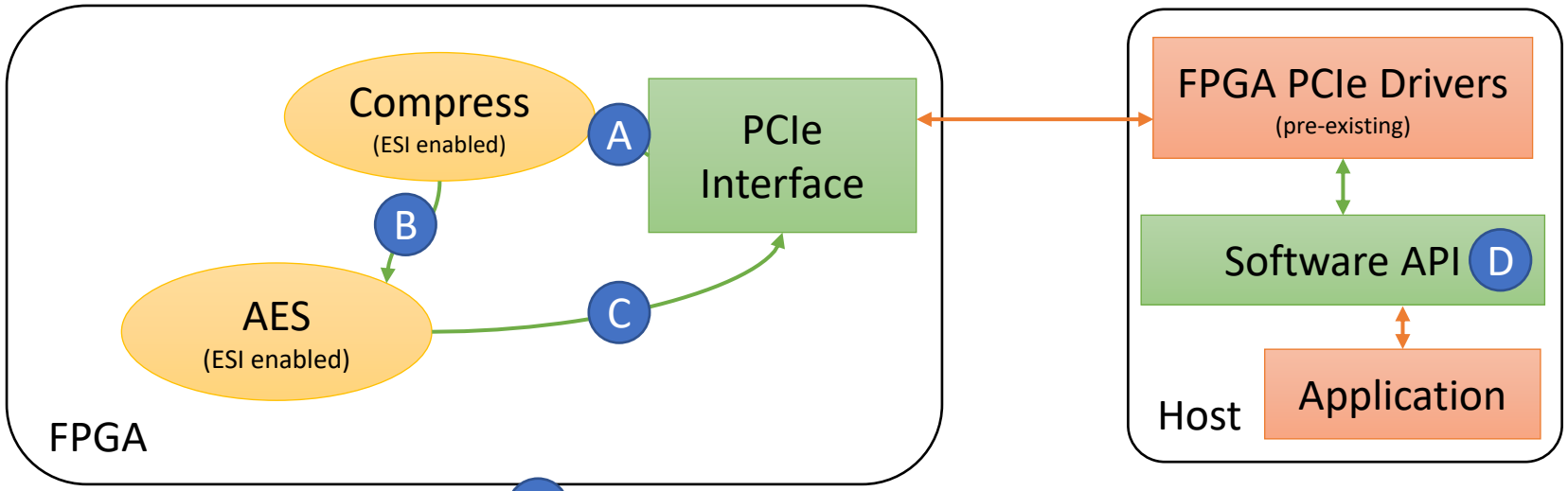
- Open Source



HLI instantiated / auto-generated

Manually coded

ESI-enabled, manually coded module



HLI instantiated / auto-generated

Manually coded

ESI-enabled, manually coded module

A

```
struct CompressEncrypt {
    uint4 compression-level;
    enum { AES128, AES 192 } encryption-type;
    enum { GCM, CBC } encryption-mode;
    byte[256] encryption-key;
    list<byte> data;
}
```

B

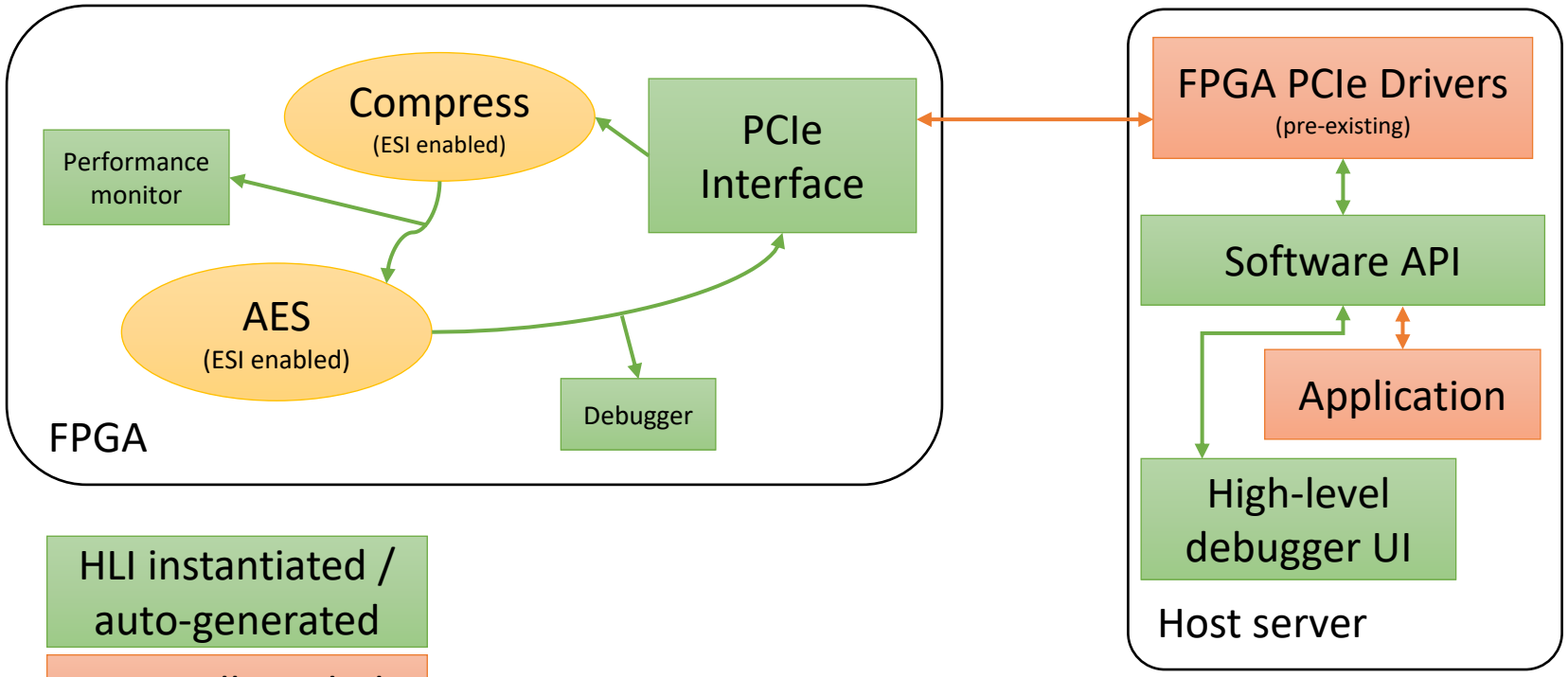
```
struct Encrypt {
    enum { AES128, AES 192 } encryption-type;
    enum { GCM, CBC } encryption-mode;
    byte[256] encryption-key;
    list<byte> compressed-data;
}
```

C

```
list<byte> compressed-encrypted-data;
```

D

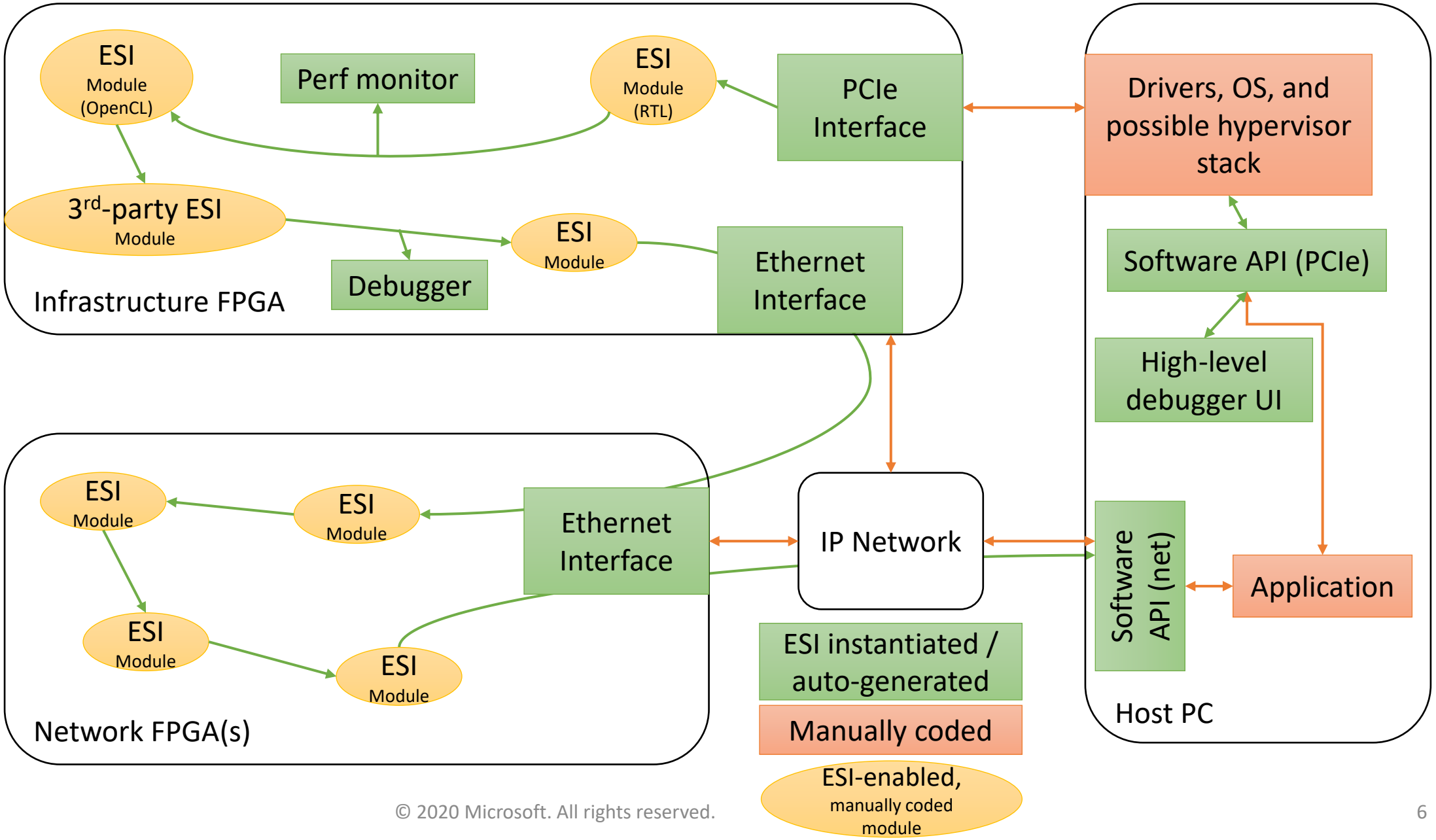
```
class CompressEncryptAccelerator {
    void SendCompressEncrypt(struct CompressEncrypt);
    void RecvData(list<byte>& compressed_encrypted_data);
}
```



HLI instantiated / auto-generated

Manually coded

ESI-enabled, manually coded module



Benefits

- Composability
- Inter-language communication
- Type checking to reduce bugs at interface boundaries
- Correct-by-construction building of communication fabric (including clock domain crossings)
- Automated decision making about the physical signaling between modules
- Automated, typed software API generation which bridges over PCIe, network, or simulation
- Automated endianness conversions
- Automatic pipelining and floor planning to reduce timing closure pressure
- Compatibility between modules with different bandwidths (automatic gearboxing)
- Type and signal aware debuggers/monitors in communication fabric
- Common interface for board support packages
- Extensible services to support global resources (e.g. telemetry)

Goals for collaboration

- Standardize hardware communication model/representation
  - My mental model is dataflow
  - This isn’t the only way to model this
- Standardize interconnect type system
  - int, bit, byte, bool
  - struct, array
  - parameterized-width floats, fixed-point
  - lists (variably-sized)
- Standardize BSP interfaces
  - ESI “services”