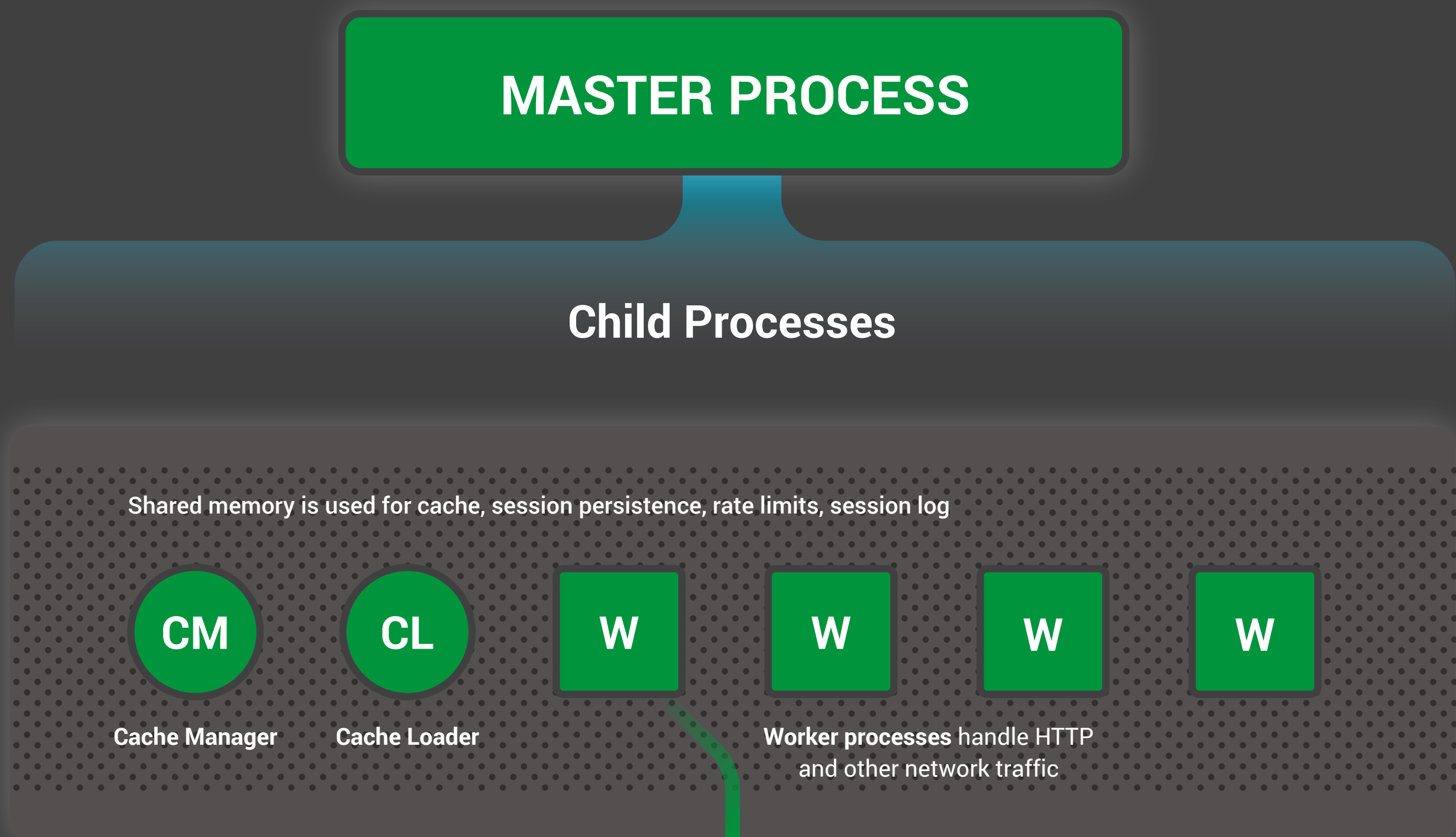
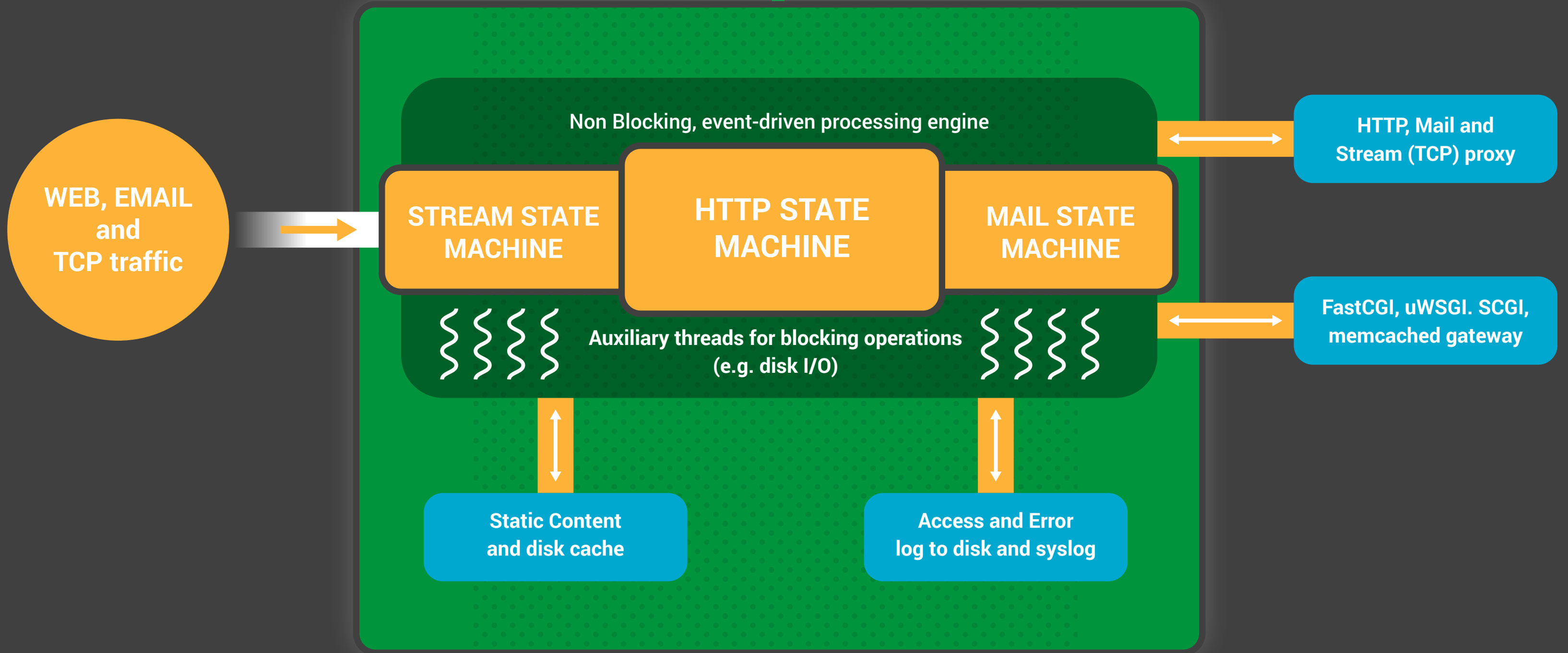


# NGINX ARCHITECTURE

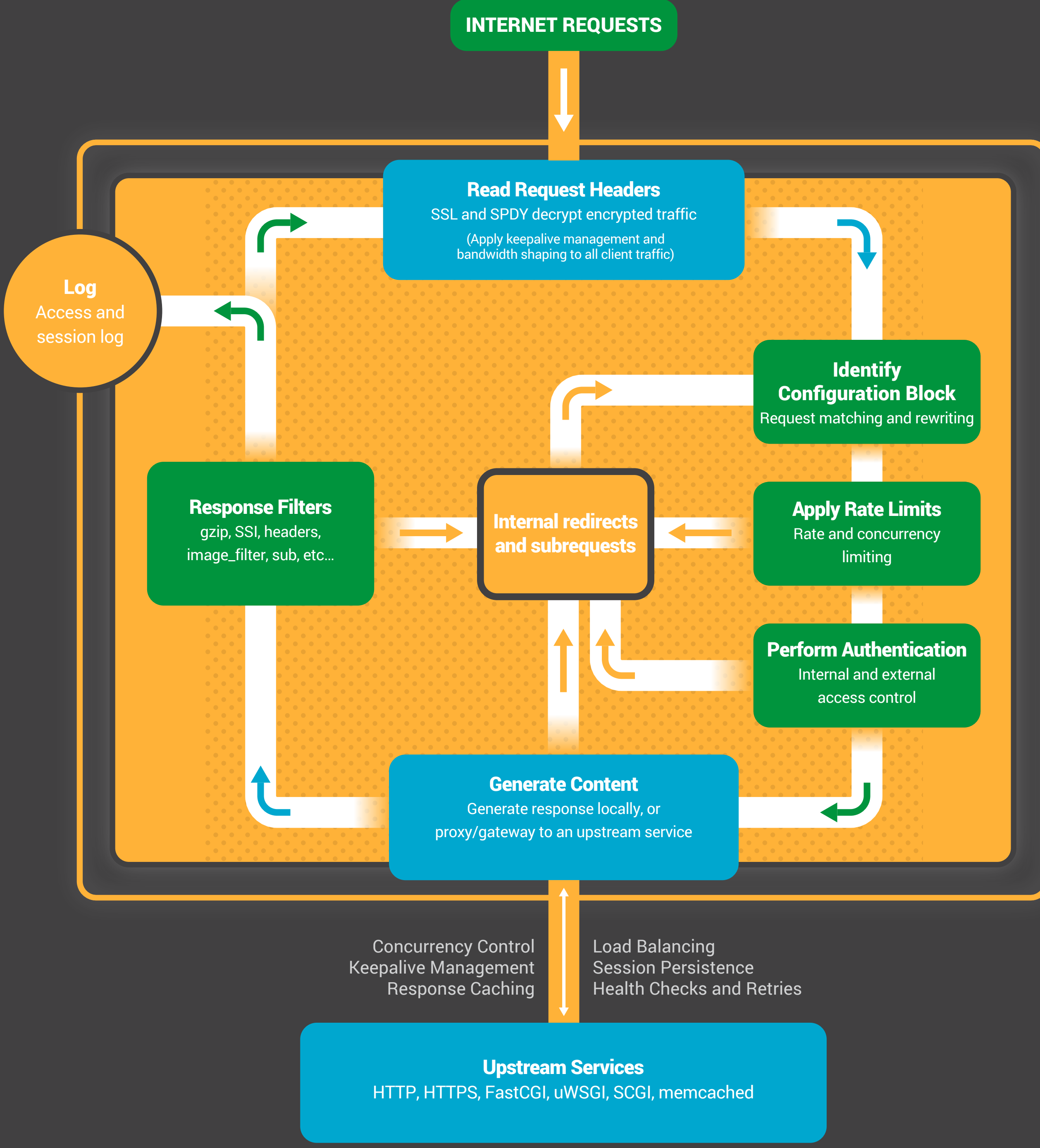
## NGINX Process Architecture



## Inside the NGINX Worker Process



## Life of an HTTP Request



## BLOCKING AND NON-BLOCKING I/O

### Most web application platforms use blocking (waiting) I/O

Listen Sockets (port 80, 443, etc)



- Wait for an event (epoll or kqueue)
- accept [ ] new connection socket [ ]
- read [ ] wait until request is read
- write [ ] wait until response is written
- wait [ ] wait on KeepAlive connection
- on error...
- close [ ]

Each worker can only process one active connection at a time

### NGINX uses a Non-Blocking "Event-Driven" architecture

Listen Sockets & Connection Sockets

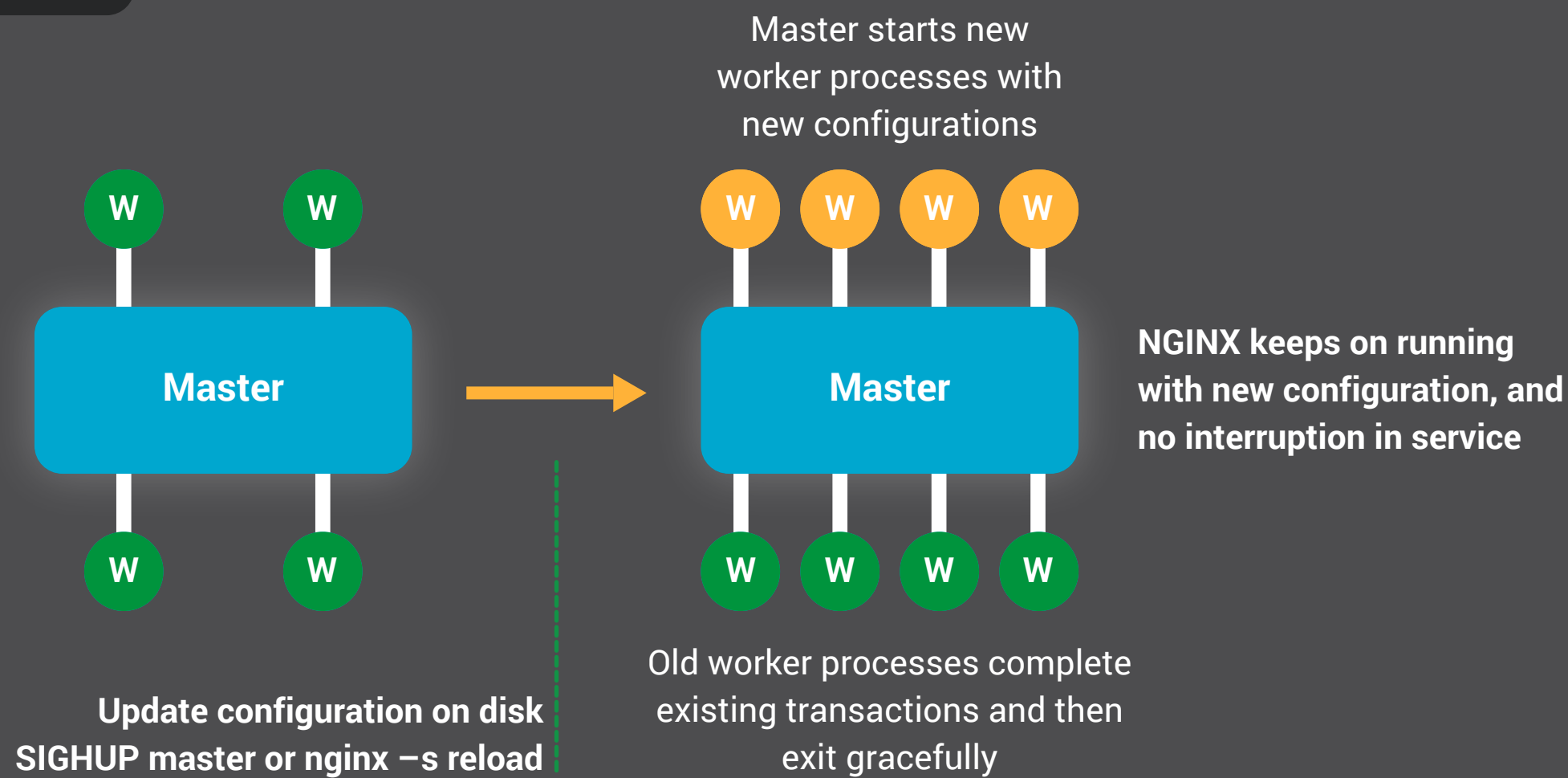


- Wait for an event (epoll or kqueue)
- Event on Listen Socket:
  - accept [ ] new [ ]
  - set [ ] to be non-blocking
  - add [ ] to the socket list
- Event on Connection Socket:
  - data in read buffer? read [ ]
  - space in write buffer? write [ ]
  - error or timeout? close [ ] & remove [ ] from socket list

An NGINX worker can process hundreds of thousands of active connections at the same time

## UPDATING CONFIGURATION, UPGRADING NGINX

### Load new configuration with no downtime



### Load new NGINX binary with no downtime

