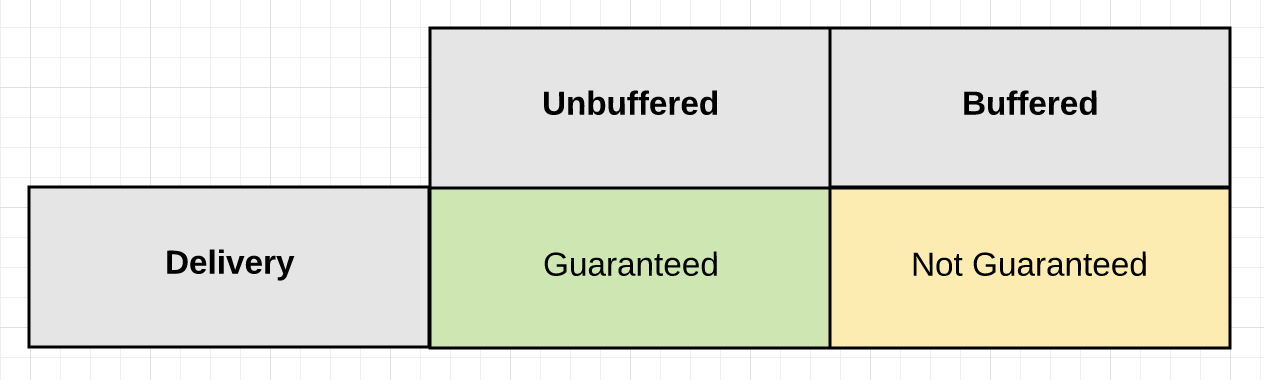
## Channels

Channels allow goroutines to communicate with each other through the use of signaling semantics. Channels accomplish this signaling through the use of sending/receiving data or by identifying state changes on individual channels. Don't architect software with the idea of channels being a queue, focus on signaling and the semantics that simplify the orchestration required.

## Diagrams

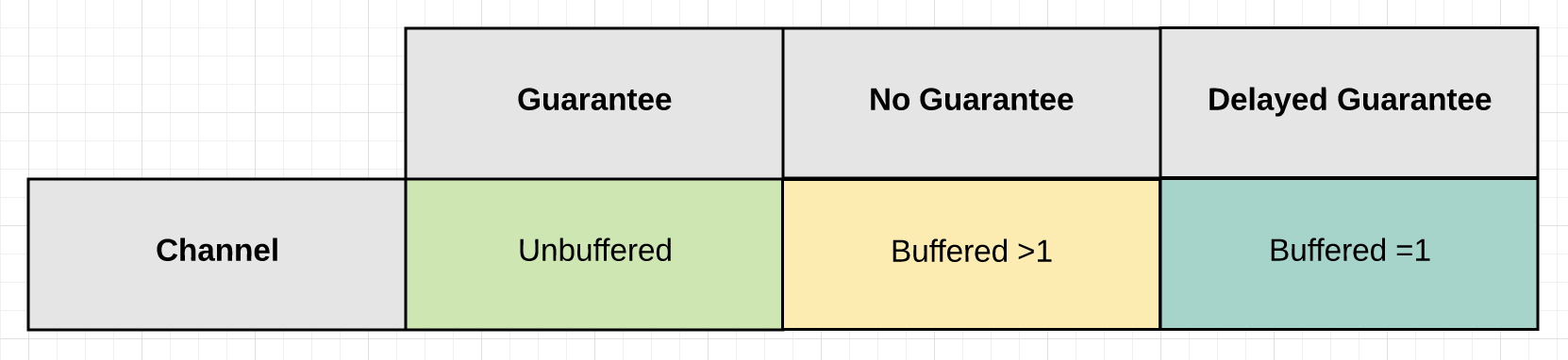
### Guarantee Of Delivery

The Guarantee Of Delivery is based on one question: “Do I need a guarantee that the signal sent by a particular goroutine has been received?”

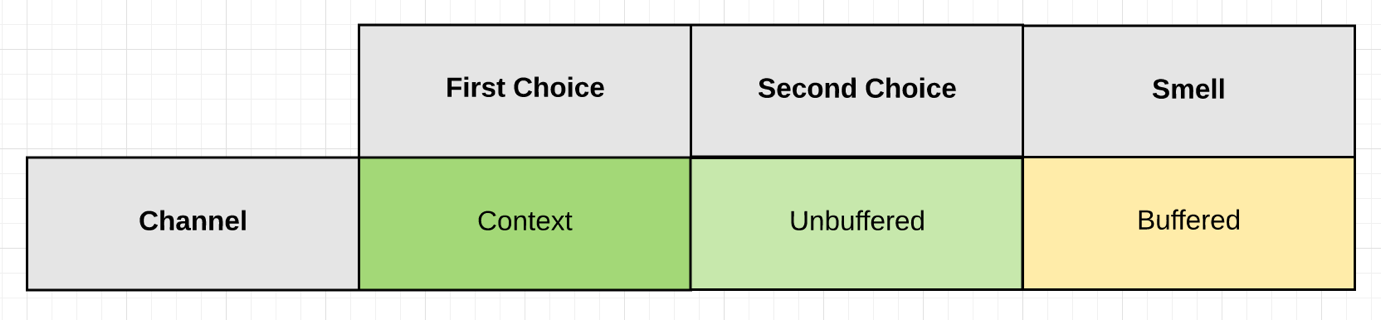
[](https://github.com/ardanlabs/gotraining/blob/master/topics/go/concurrency/channels/guarantee_of_delivery.png)

### Signaling With Or Without Data

When you are going to signal with data, there are three channel configuration options you can choose depending on the type of guarantee you need.

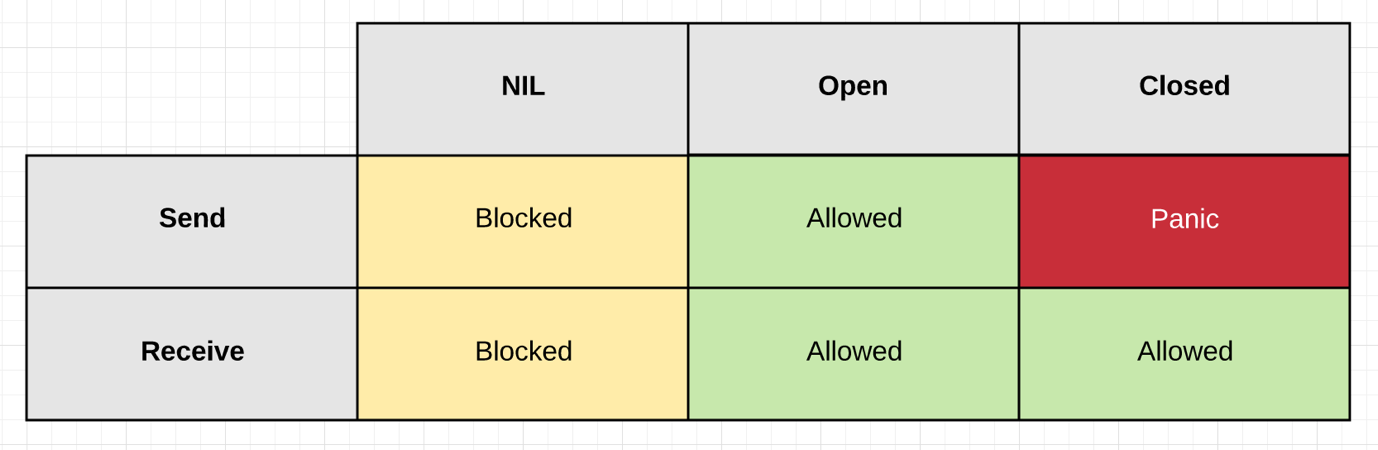
[](https://github.com/ardanlabs/gotraining/blob/master/topics/go/concurrency/channels/signaling_with_data.png)

Signaling without data serves the main purpose of cancellation. It allows one goroutine to signal another goroutine to cancel what they are doing and move on. Cancellation can be implemented using both unbuffered and buffered channels.

[](https://github.com/ardanlabs/gotraining/blob/master/topics/go/concurrency/channels/signaling_without_data.png)

### State

The behavior of a channel is directly influenced by its current State. The state of a channel can be nil, open or closed.

[](https://github.com/ardanlabs/gotraining/blob/master/topics/go/concurrency/channels/state.png)

## Links

[The Behavior Of Channels](https://www.ardanlabs.com/blog/2017/10/the-behavior-of-channels.html)   
[The Nature Of Channels In Go](https://www.ardanlabs.com/blog/2014/02/the-nature-of-channels-in-go.html)

## Buffer Bloat - 2011

* Large buffers prevent timely notification of back pressure.
* They defeat your ability to reduce back pressure in a timely matter.
* They can increase latency not reduce it.
* Use buffered channels to provide a way of maintaining continuity.
  + Don't use them just for performance.
  + Use them to handle well defined bursts of data.
  + Use them to deal with speed of light issues between handoffs.

[Bufferbloat: Dark Buffers in the Internet](https://www.youtube.com/watch?v=qbIozKVz73g)  
[Buffer Bloat Videos](http://www.bufferbloat.net/projects/cerowrt/wiki/Bloat-videos)