

# WebRTC ICE & Peer State

Cullen & Justin  
October 2012

# Basic Idea for ICE State

- Separate gathering and checking states
- Have checking state only represent aggregate information likely to be useful to application



# ICE Gathering States

- **RTCGatheringState Enum**

- new

The object was just created, and no networking has occurred yet.

- gathering

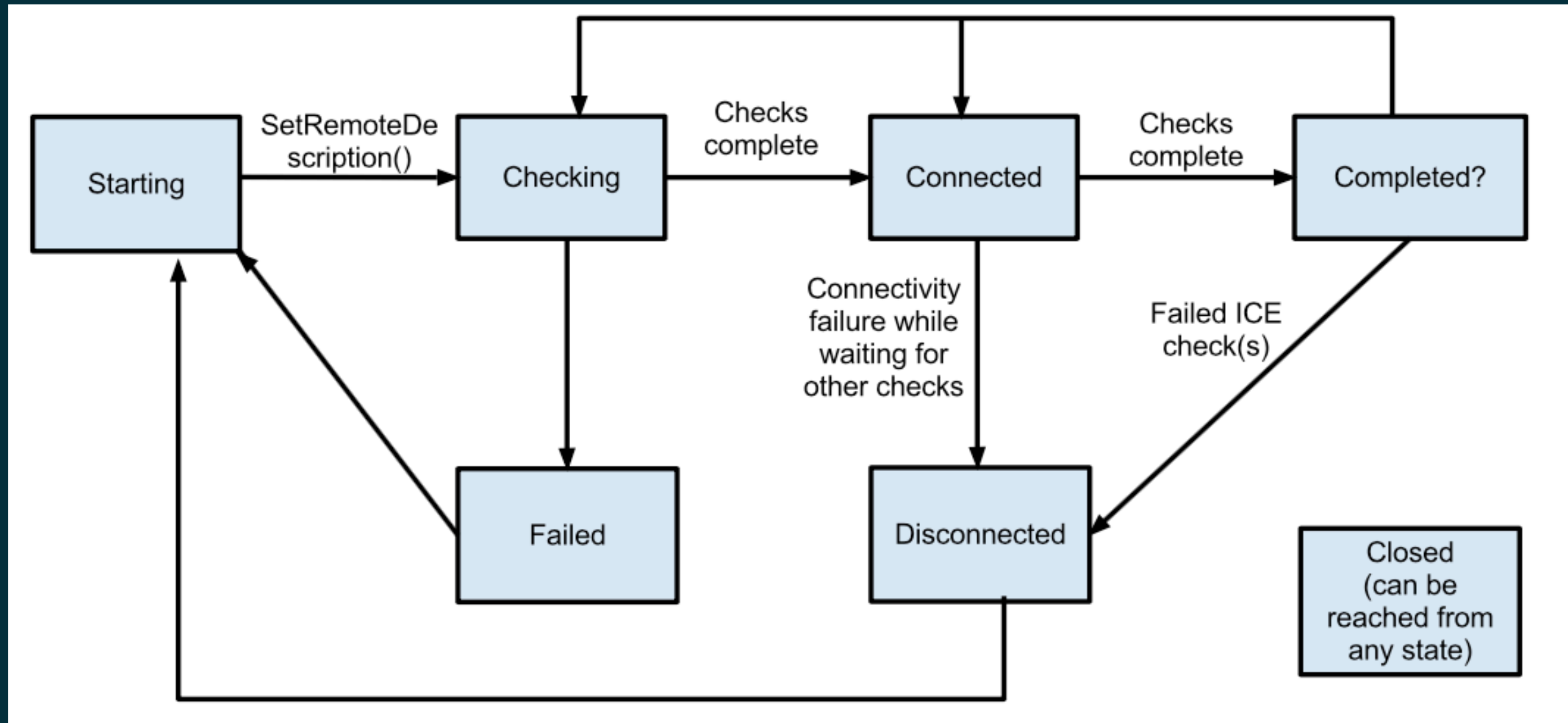
The ICE engine is in the process of gathering candidates for this RTCPeerConnection.

- complete

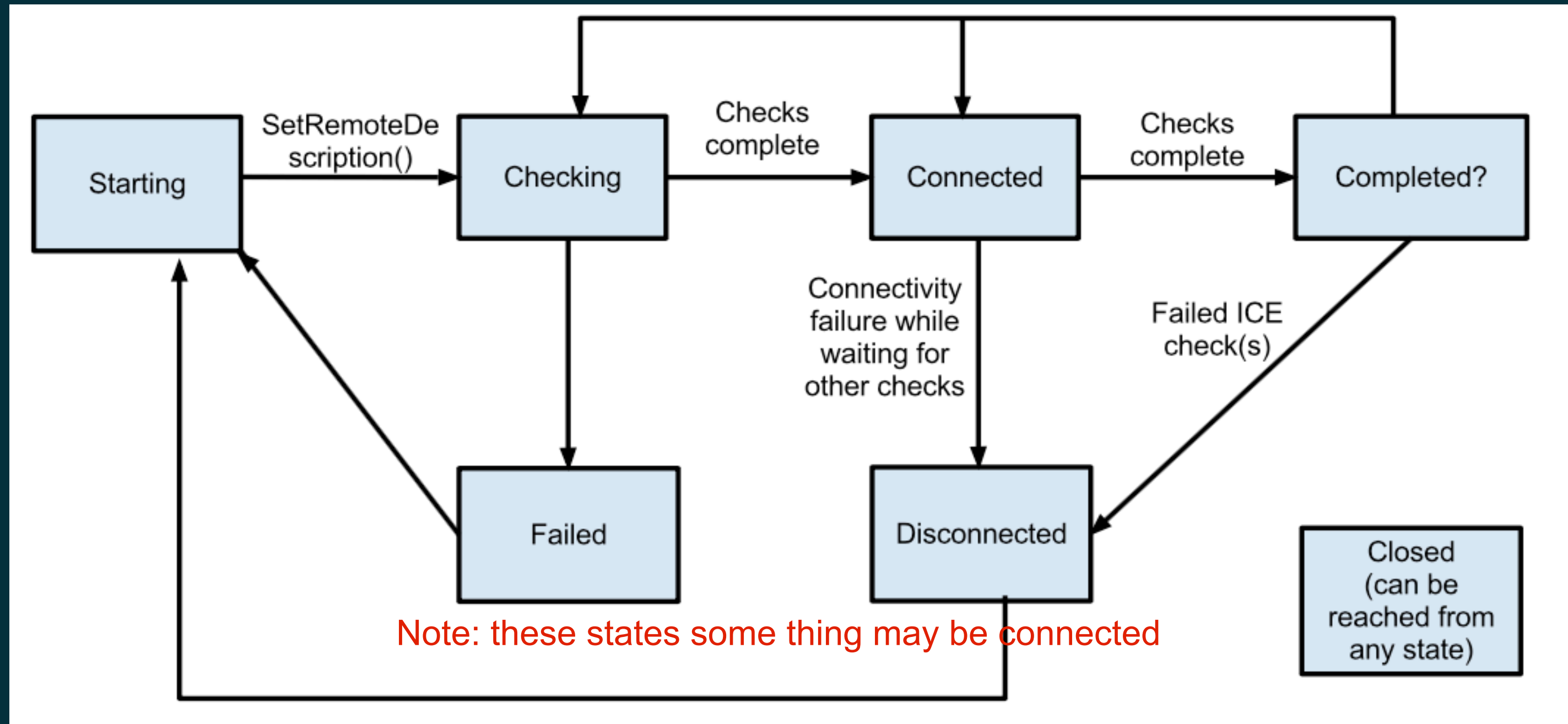
The ICE engine has completed gathering. Events such as adding a new interface or new TURN server could cause the state to go back to gathering.



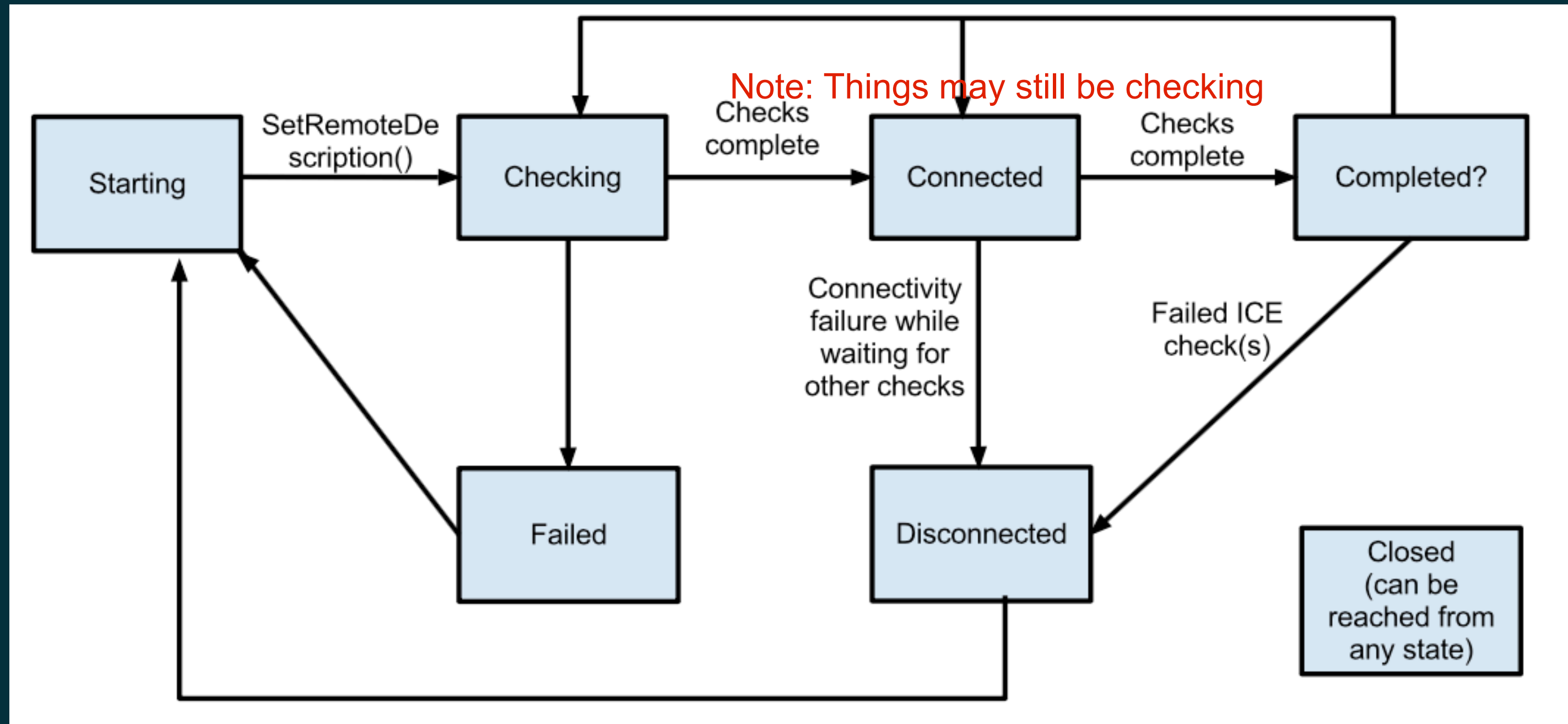
# ICE Checking States



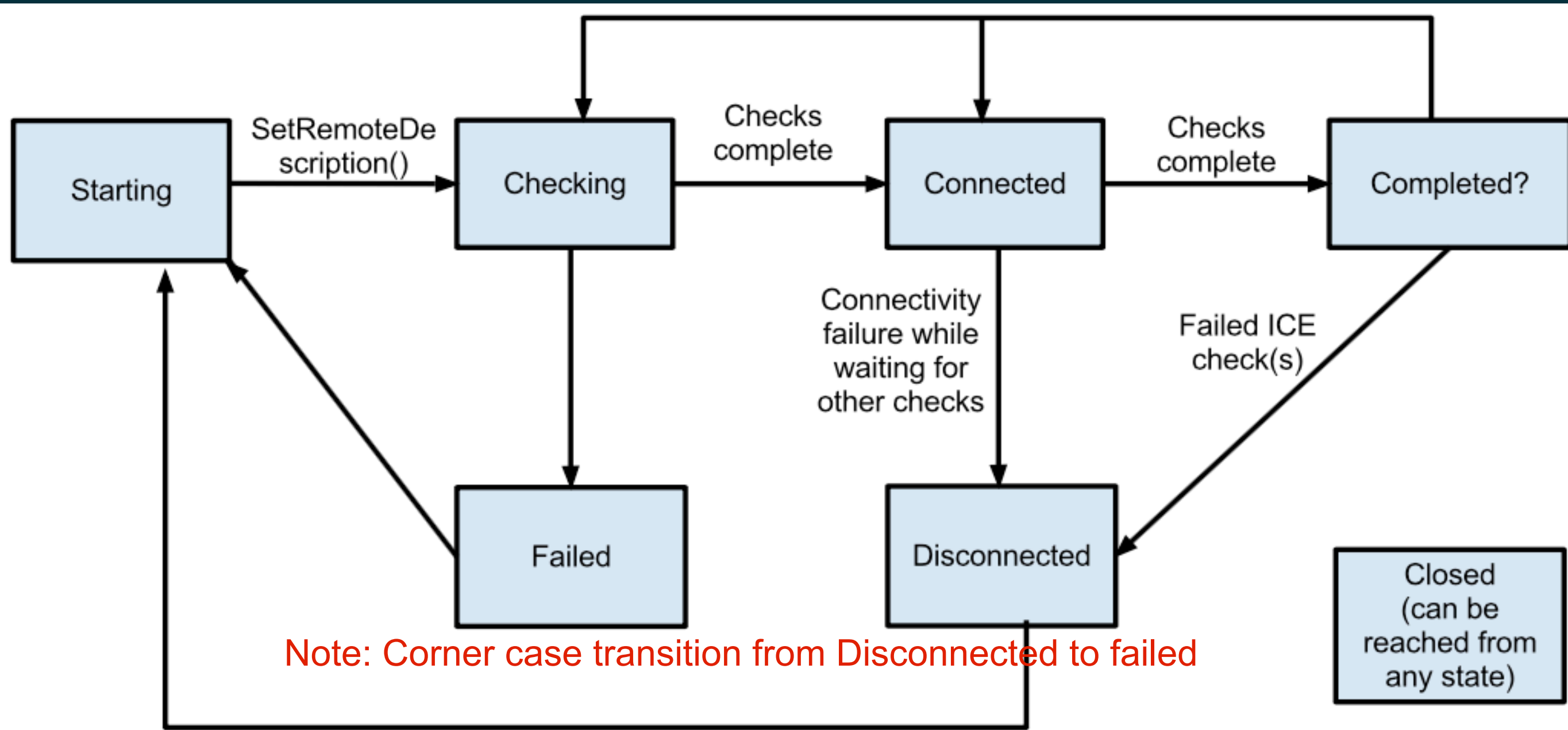
# ICE Checking States



# ICE Checking States



# ICE Checking States



# Proposed Change of state name

- Rename “starting” to “new”
- State name accessor should be
  - iceConnectState
  - iceGatheringState
- change names of data types to match
- State change callbacks should be one of (based on later slides of 1 or 2 callbacks)
- if 2 callbacks
  - oniceconnectchange and onicegatheringchange
- if 1 callback
  - onicechange



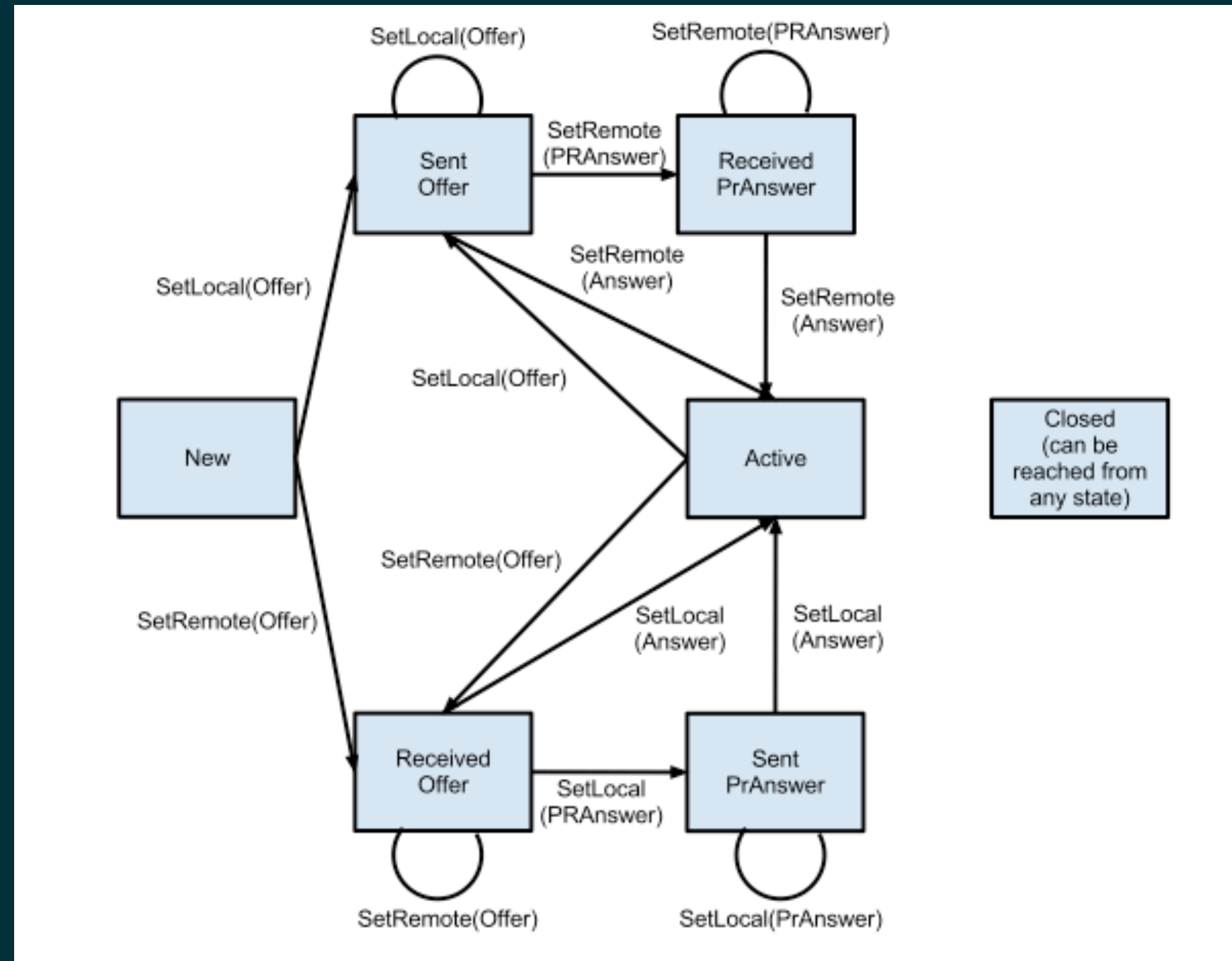


# Proposed change ICE Callback

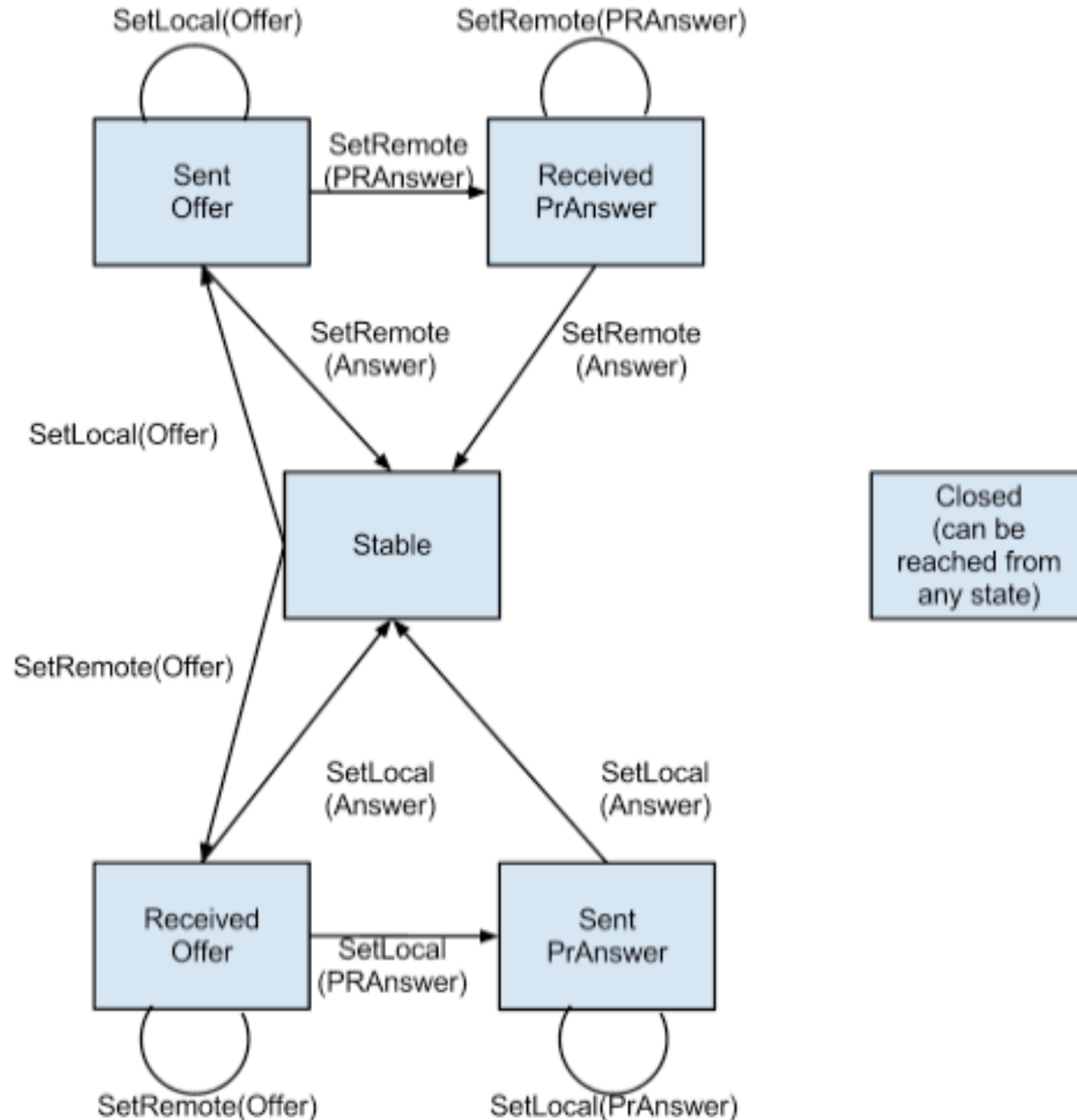
- Right now when gathering completes, get two callbacks
  - ongatheringchange
  - onicecandidate with null candidate
- Two proposals to sort this out
  - A: Keep both callback but don't have the final null candidate callback onicecandidate
  - B: remove the ongatheringchange callback
- We are suggesting B



# Basic Idea for Peer State



# Proposed Change for Peer State



- Merge New + Active into new state called Stable