



# CHAPTER 5

## Sequence Diagrams for Components

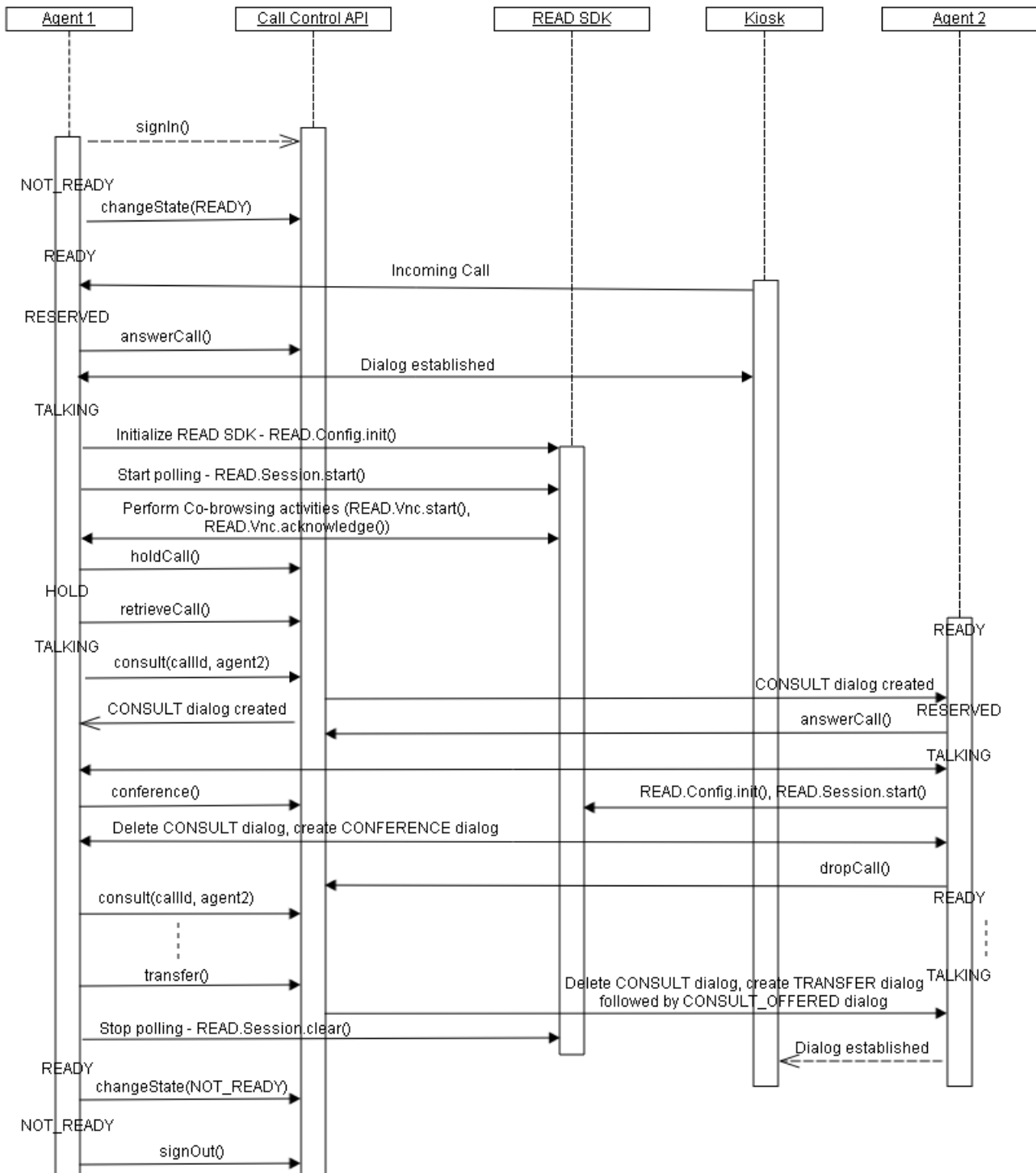
---

### API in Possible Use Case Scenarios

The APIs support the following capabilities:

- User Sign In/Sign Out
- Change Agent States
- Call Control
- Event Notification (call back handler to clients that subscribe to that class of resource)

Figure 4 Detailed Communication of Call Control Scenario



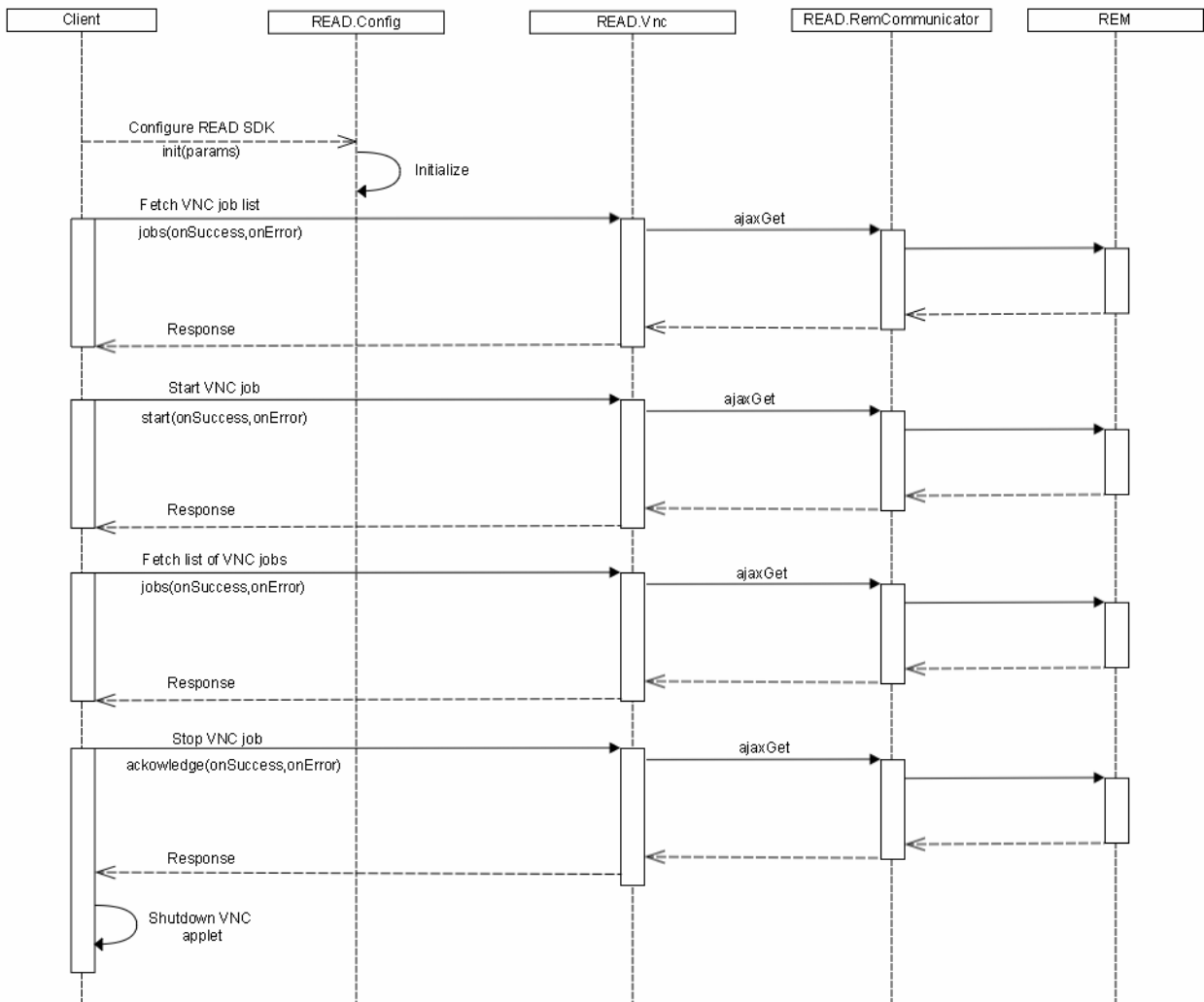
Supported use:

- Create a call (dialog) between VTM and VTA (agent)
- Put active call on hold
- Consult another remote agent and followed by conference or transfer the call

## VNC Module

This section contains detail communication diagrams of VNC module.

**Figure 5 Detailed Communication of VNC Module**



## Example Usage of READ SDK

This section provides some sample codes for using READ SDK in client application.

### READ SDK Recommended Practices

- Implement the `onError` callback while configuring READ SDK through `READ.Config.init()` method. This will give an opportunity to handle accidental drop of REM session or HTTP connection:

```
READ.Config.init({
  onError: function(error) {
    switch(error.status) {
      case 0:
      case 12029:
        console.error('Communication Failure.');
```

```
        break;
      case 420:
        var details = error.details;
        var msg = details.message;
        var code = details.code;
        console.error('Server reported an error. ', code, msg);
        if (code === READ.Constants.ErrorCode.NO_REM_SESSION) {
          READ.Session.clear();
        }
        break;
    }
  }.
});
```

- Take care to verify the correct URL is provided as `codebase` property while creating the `APPLET` element through the `createAppletSource()` method of `VNC` class.
- Stop Co-browsing job from the UI rather than disconnecting from VNC client.

### Reference Implementation of VNC feature

- Define a placeholder element for the VNC co-browsing client applet:

```
<div id="appletplace"> </div>
```

- Start the Job at REM Server:

```
READ.Vnc.start(function(data) {
  console.info('Vnc request sent successfully');
}, function(err, response) {
  alert('Failed to start VNC. Status: ' + error.status + ', Cause: ' +
  error.statusText);
});
```

- Poll for Job Status to Handle State of Application:

```
READ.Event.subscribe(READ.Session.Topic.REM_AGENT_SESSION, function(isActive) {
  if(!isActive) {
    if(READ.Vnc.isActive()) {
      READ.Vnc.acknowledge();
    }
    document.getElementById('appletplace').innerHTML = "";
    disableStartVncBtn();
  }
});
```

```

disableStopVncBtn();
}
});
READ.Event.subscribe(READ.Vnc.Topic.VNC_EMPTY, function() {
enableStartVncBtn();
disableStopVncBtn();
});
READ.Event.subscribe(READ.Vnc.Topic.VNC_INITED, function(job) {
disableStartVncBtn();
if(job.Owner === READ.Config.agent) {
enableStopVncBtn();
} else {
disableStopVncBtn();
}
});
READ.Event.subscribe(READ.Vnc.Topic.VNC_COMPLETED, function(job) {
var appletPlace = document.getElementById('appletplace').innerHTML;
if(job.Owner === READ.Config.agent) {
disableStartVncBtn();
enableStopVncBtn();
// If applet is not already loaded, create an invisible
// APPLET by setting the width and height as 1px.
if(!appletPlace) {
var appletsource = READ.Vnc.createAppletSource( {width:1, height:1}, {codebase:
window.location.origin +'/FinesseAPI/readsdk', iecIp: job.IecIp,
iecPassword:job.IecPasswd, vncServerPort:job.VncServerPort } );
document.getElementById('appletplace').innerHTML = appletsource;
}
} else if(job.Owner !== READ.Config.agent) { // Probably a conference call
document.getElementById('appletplace').innerHTML = "";
disableStartVncBtn();
disableStopVncBtn();
}
});
READ.Event.subscribe(READ.Vnc.Topic.VNC_ACKNOWLEDGED, function(job) {
document.getElementById('appletplace').innerHTML = "";
enableStartVncBtn();
disableStopVncBtn();
});

```

- Acknowledge to Terminate the Job at REM:

```

READ.Vnc.acknowledge(function() {
document.getElementById('appletplace').innerHTML = "";
}, function(error) {
alert("Co-browsing stop failed. Reason: " + error.statusText);
});

```

