

FIG. 1A

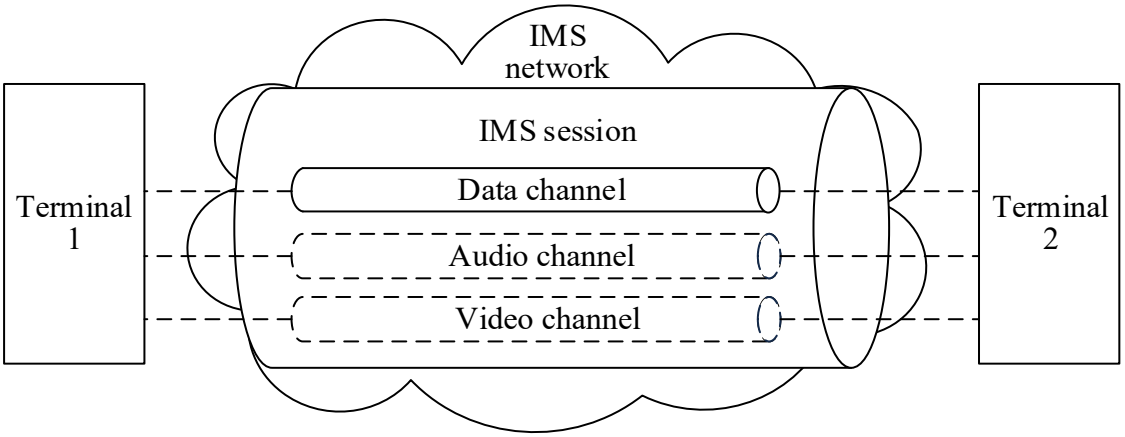


FIG. 1B

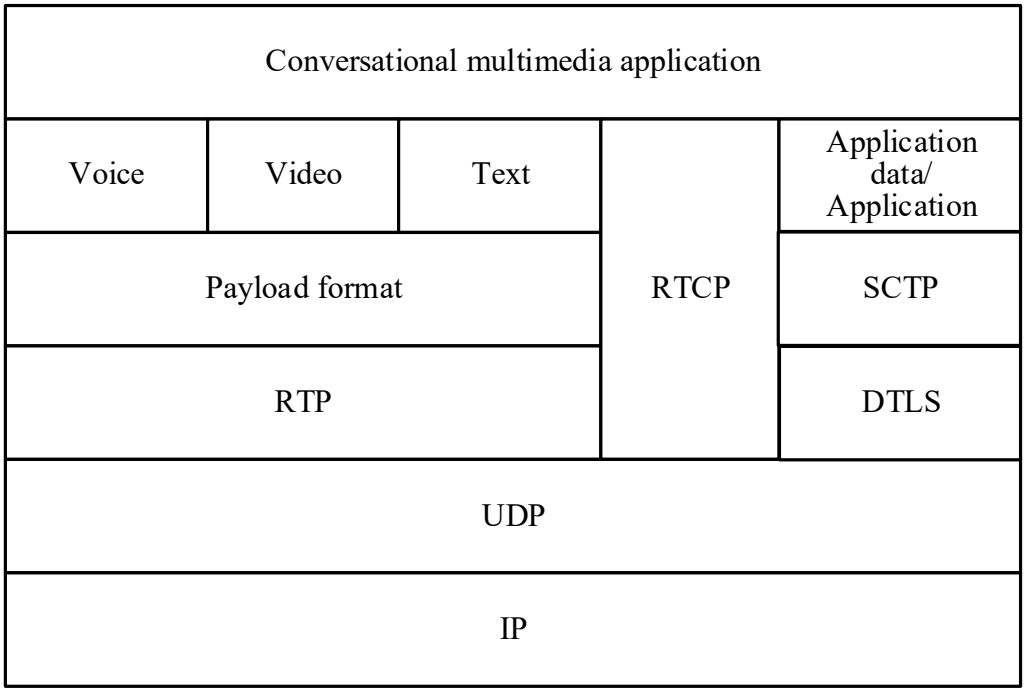


FIG. 1C

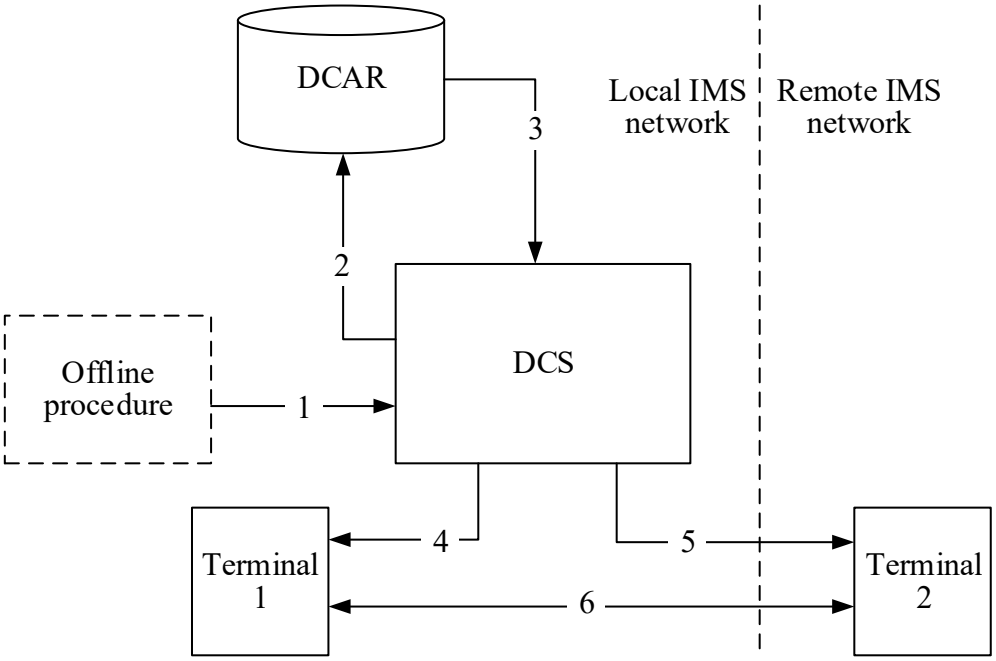

DCAR
3
Local IMS network
Remote IMS network
2
DCS
Offline procedure
1
4
5
Terminal 1
6
Terminal 2

FIG. 1D

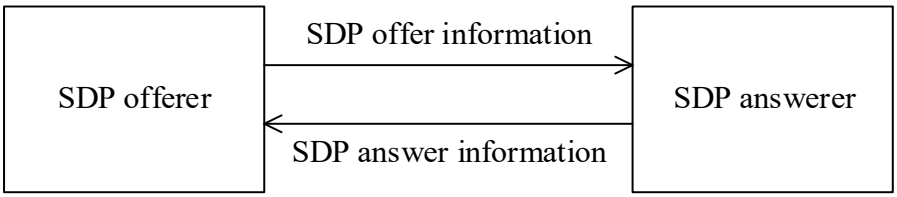

SDP offer information
SDP offerer
SDP answerer
SDP answer information

FIG. 1E

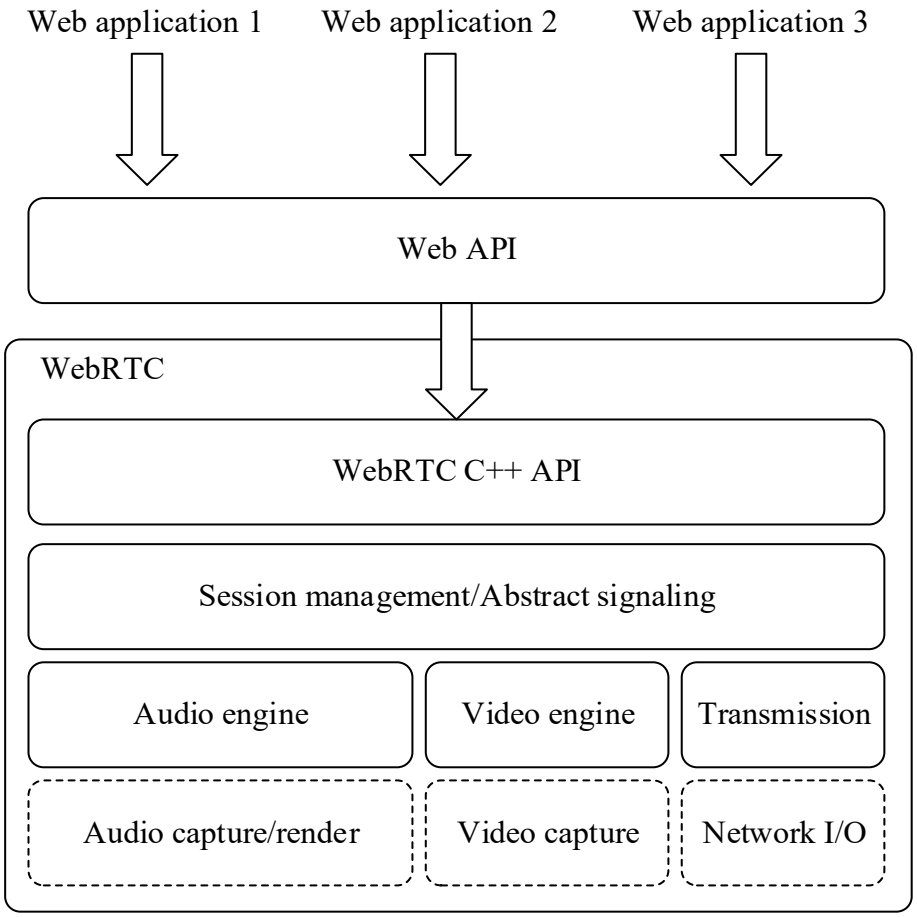

Web application 1
Web application 2
Web application 3
Web API
WebRTC
WebRTC C++ API
Session management/Abstract signaling
Audio engine
Video engine
Transmission
Audio capture/render
Video capture
Network I/O

FIG. 1F

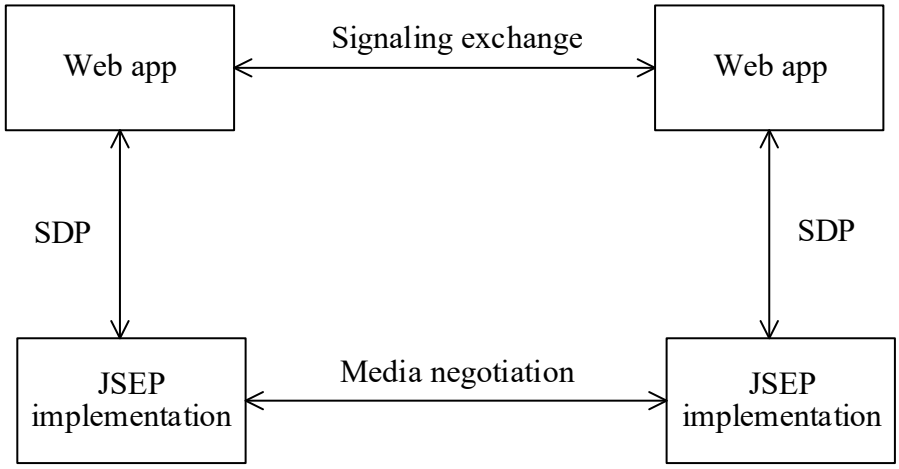

Web app
Signaling exchange
Web app
SDP
SDP
JSEP implementation
Media negotiation
JSEP implementation

FIG. 1G

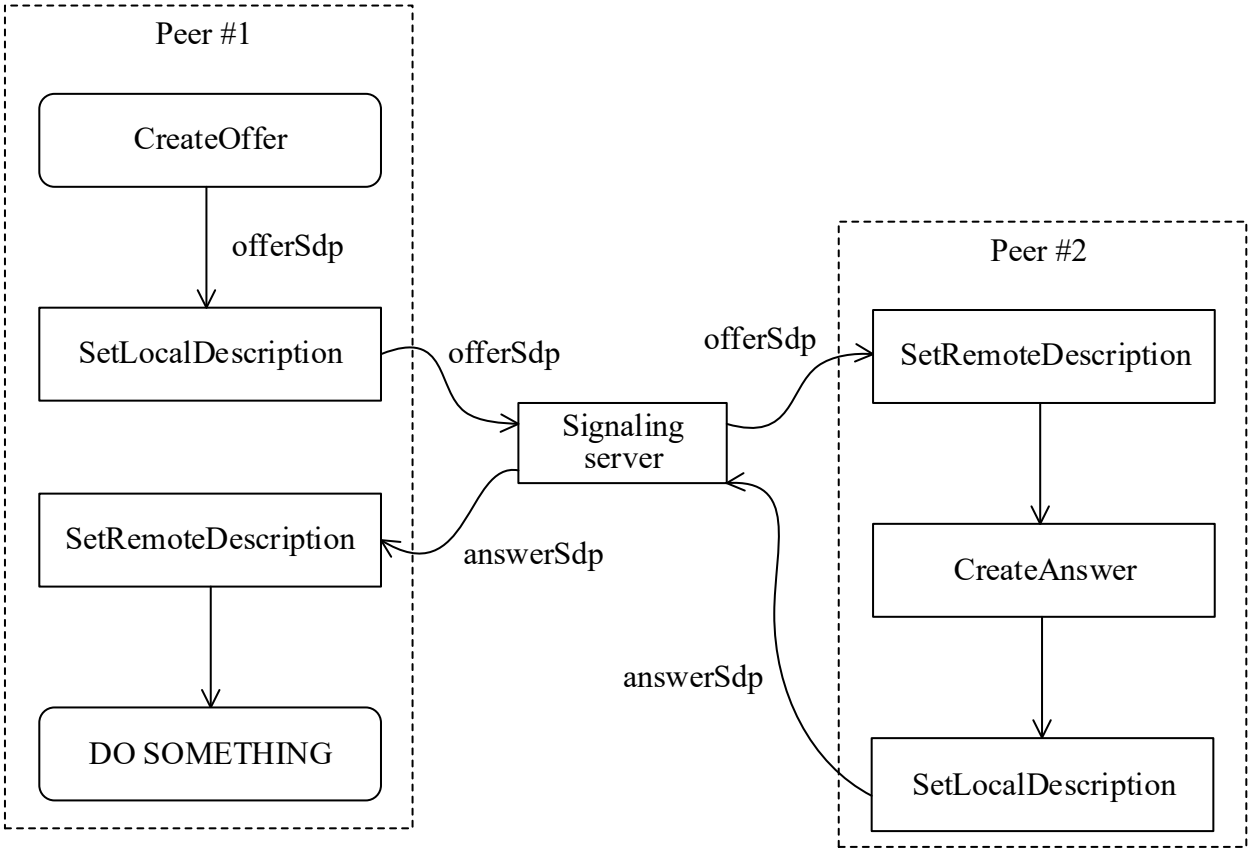

Peer #1
CreateOffer
offerSdp
SetLocalDescription
SetRemoteDescription
DO SOMETHING
offerSdp
answerSdp
Signaling server
offerSdp
answerSdp
Peer #2
SetRemoteDescription
CreateAnswer
SetLocalDescription

FIG. 1H

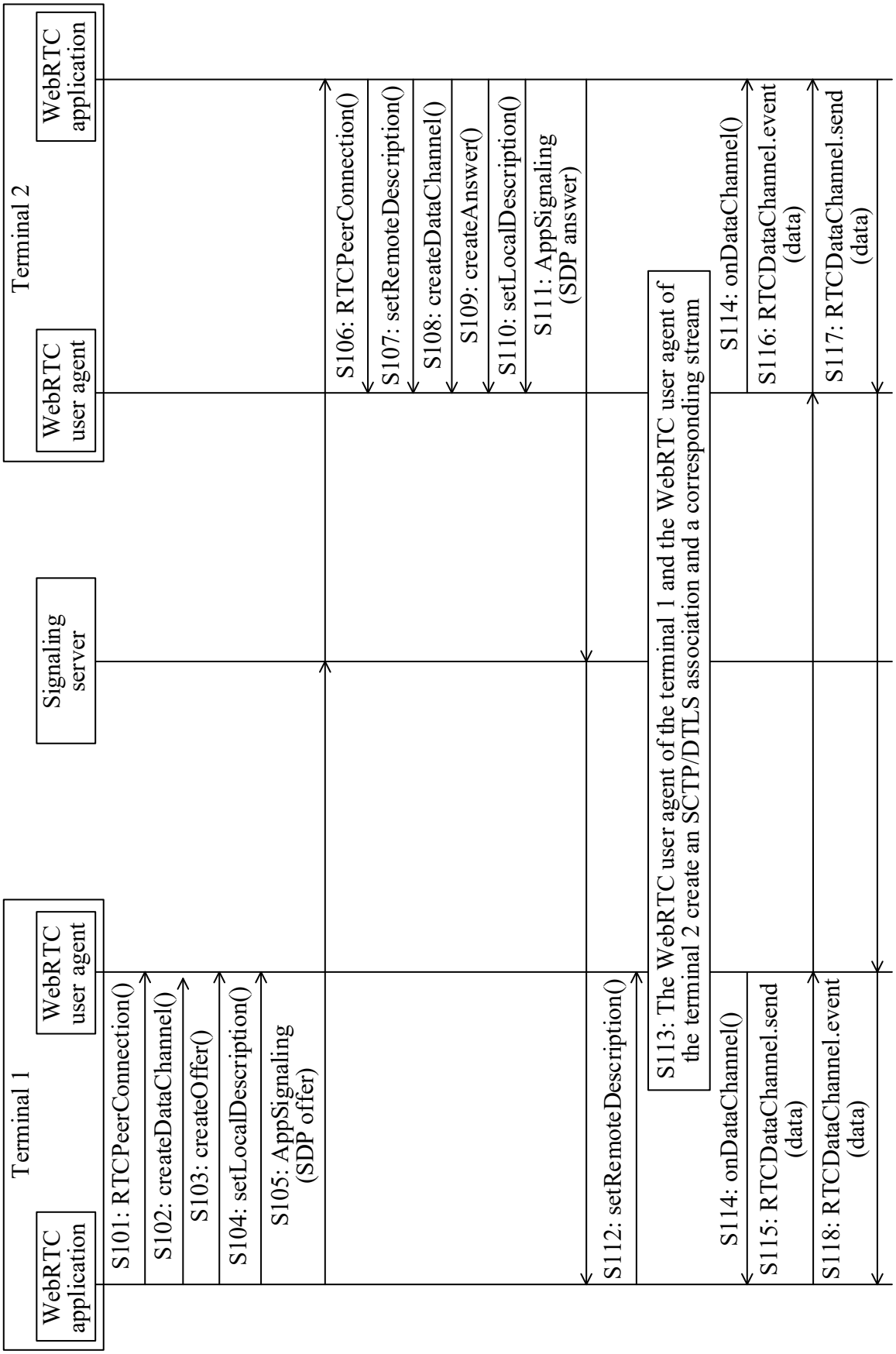

Terminal 1
WebRTC application
WebRTC user agent
Signaling server
Terminal 2
WebRTC user agent
WebRTC application
S101: RTCPeerConnection()
S102: createDataChannel()
S103: createOffer()
S104: setLocalDescription()
S105: AppSignaling (SDP offer)
S106: RTCPeerConnection()
S107: setRemoteDescription()
S108: createDataChannel()
S109: createAnswer()
S110: setLocalDescription()
S111: AppSignaling (SDP answer)
S112: setRemoteDescription()
S113: The WebRTC user agent of the terminal 1 and the WebRTC user agent of the terminal 2 create an SCTP/DTLS association and a corresponding stream
S114: onDataChannel()
S114: onDataChannel()
S115: RTCDataChannel.send (data)
S116: RTCDataChannel.event (data)
S118: RTCDataChannel.event (data)
S117: RTCDataChannel.send (data)

FIG. 1I

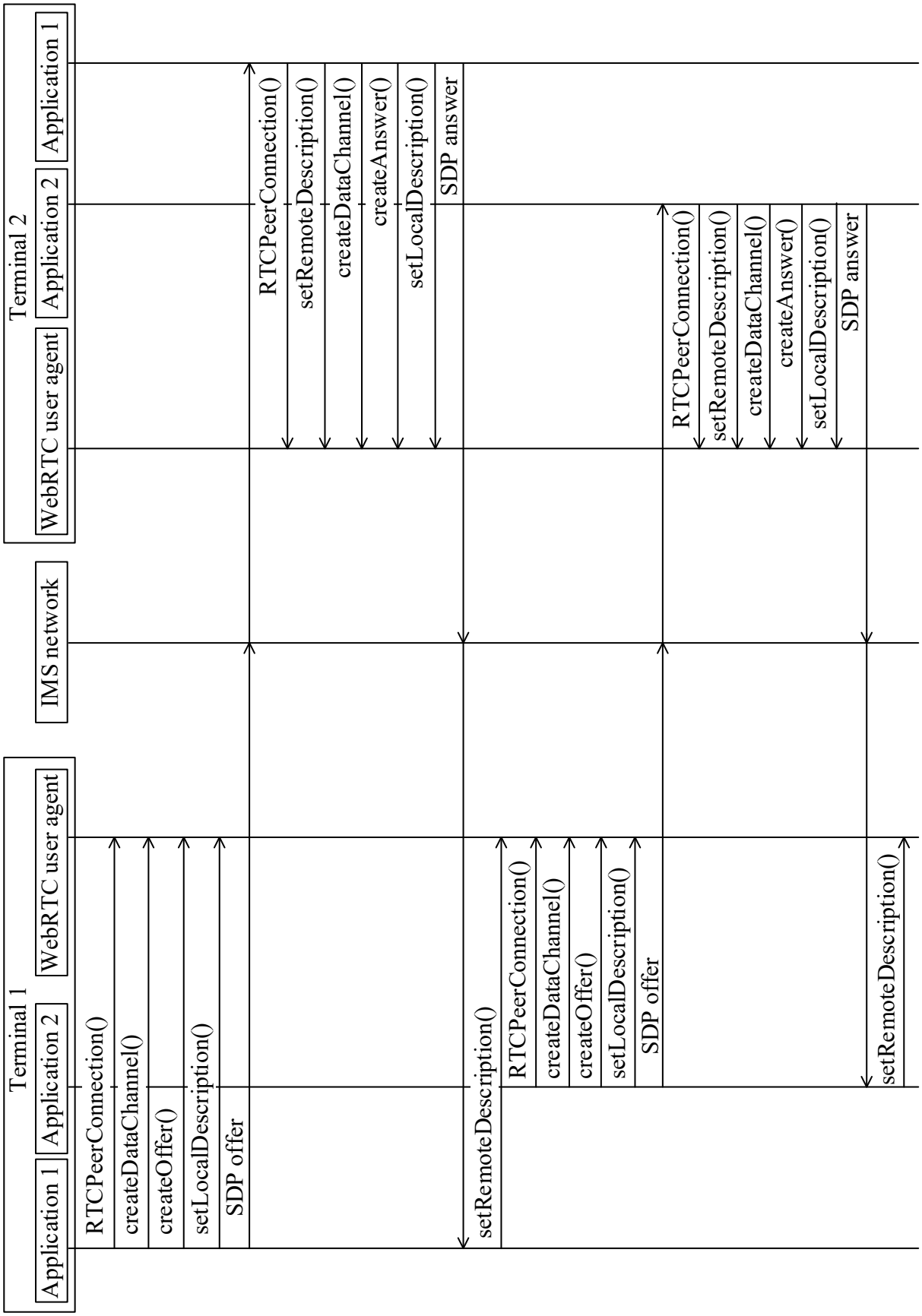

Terminal 1
Application 1
Application 2
WebRTC user agent
IMS network
Terminal 2
WebRTC user agent
Application 2
Application 1
RTCPeerConnection()
createDataChannel()
createOffer()
setLocalDescription()
SDP offer
setRemoteDescription()
RTCPeerConnection()
createDataChannel()
createOffer()
setLocalDescription()
SDP offer
RTCPeerConnection()
setRemoteDescription()
createDataChannel()
createAnswer()
setLocalDescription()
SDP answer
setRemoteDescription()
RTCPeerConnection()
setRemoteDescription()
createDataChannel()
createAnswer()
setLocalDescription()
SDP answer

FIG. 2

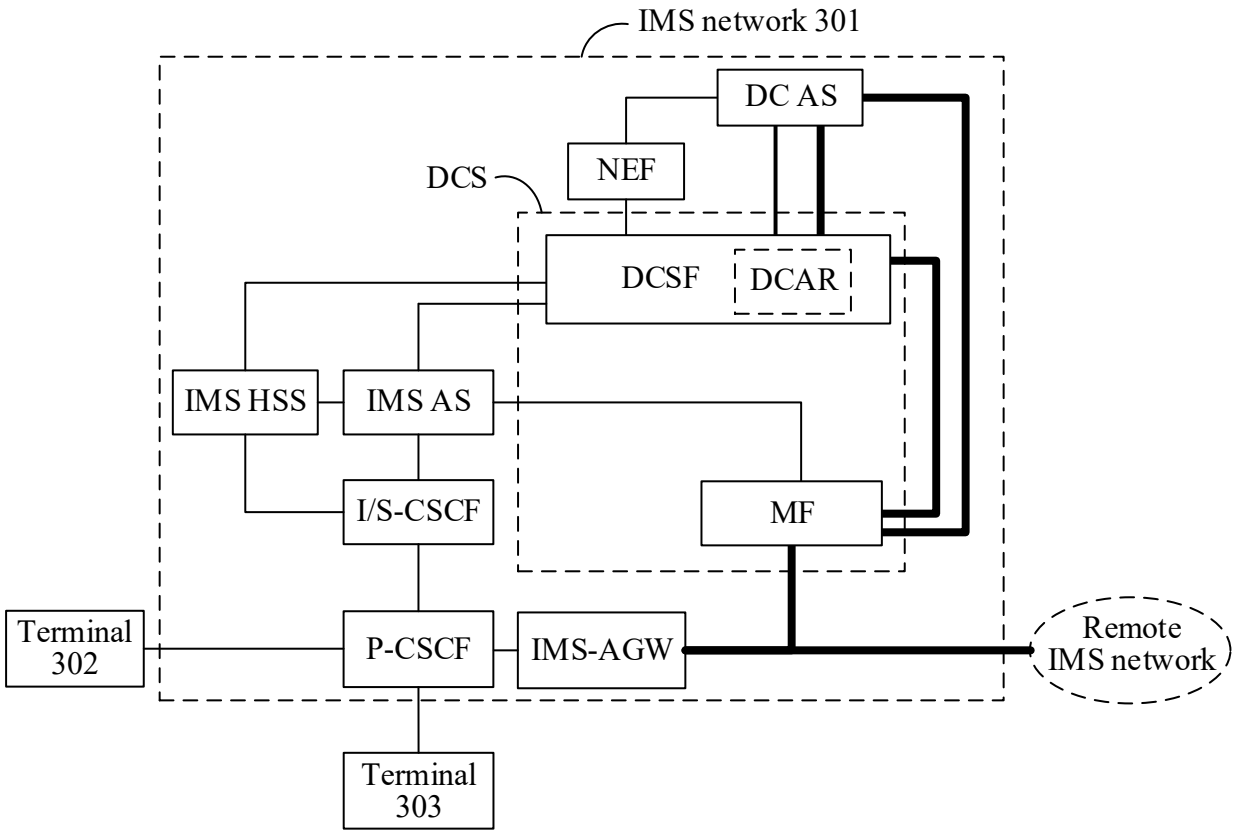

IMS network 301
DC AS
NEF
DCS
DCSF
DCAR
IMS HSS
IMS AS
I/S-CSCF
MF
Terminal 302
P-CSCF
IMS-AGW
Remote IMS network
Terminal 303

FIG. 3A

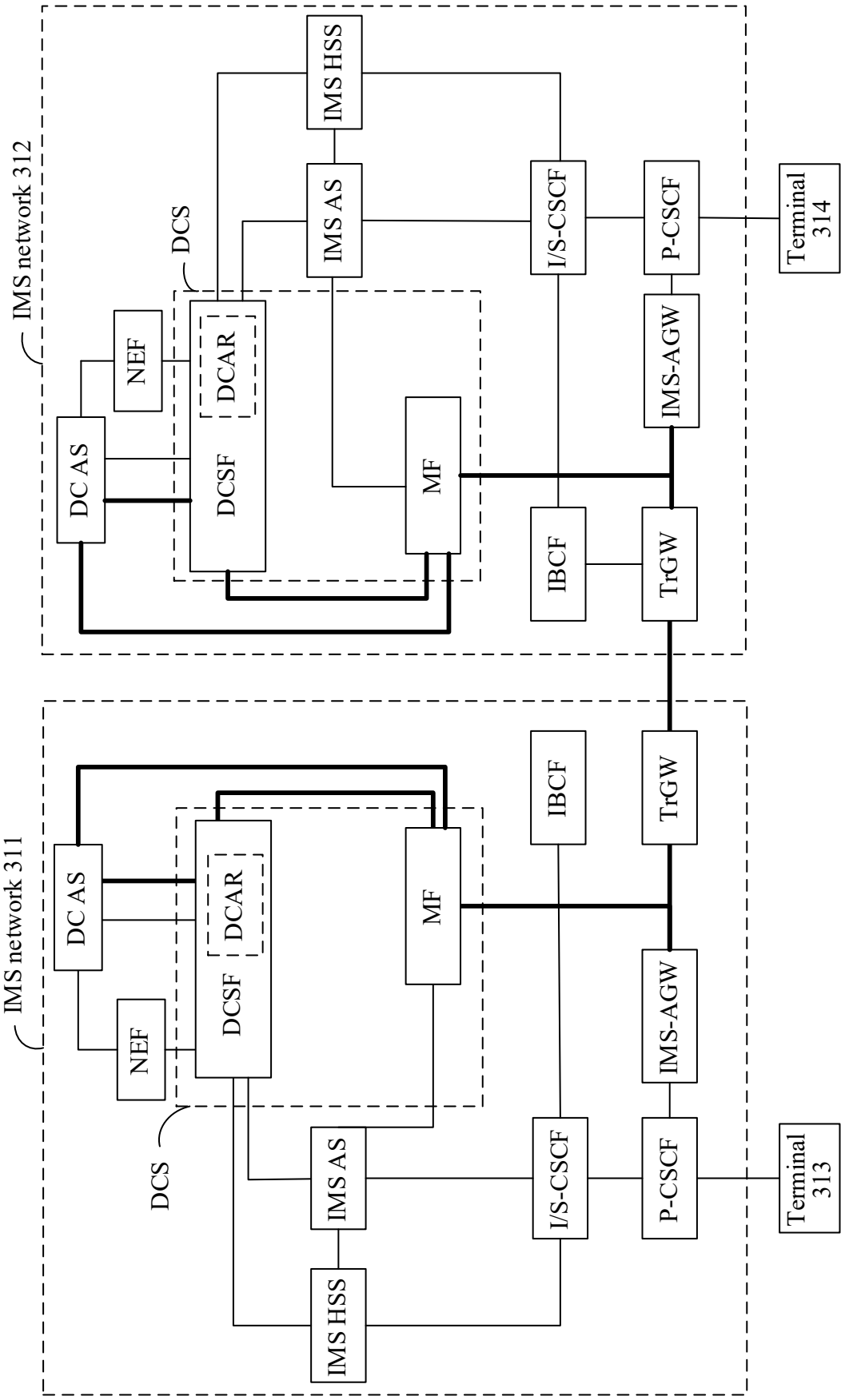

IMS network 312
IMS HSS
IMS AS
I/S-CSCF
P-CSCF
Terminal 314
DCS
NEF
DCAR
DCSF
DC AS
MF
IMS-AGW
IBCF
TrGW
IMS network 311
IMS HSS
IMS AS
I/S-CSCF
P-CSCF
Terminal 313
DCS
NEF
DCAR
DCSF
DC AS
MF
IMS-AGW
IBCF
TrGW

FIG. 3B

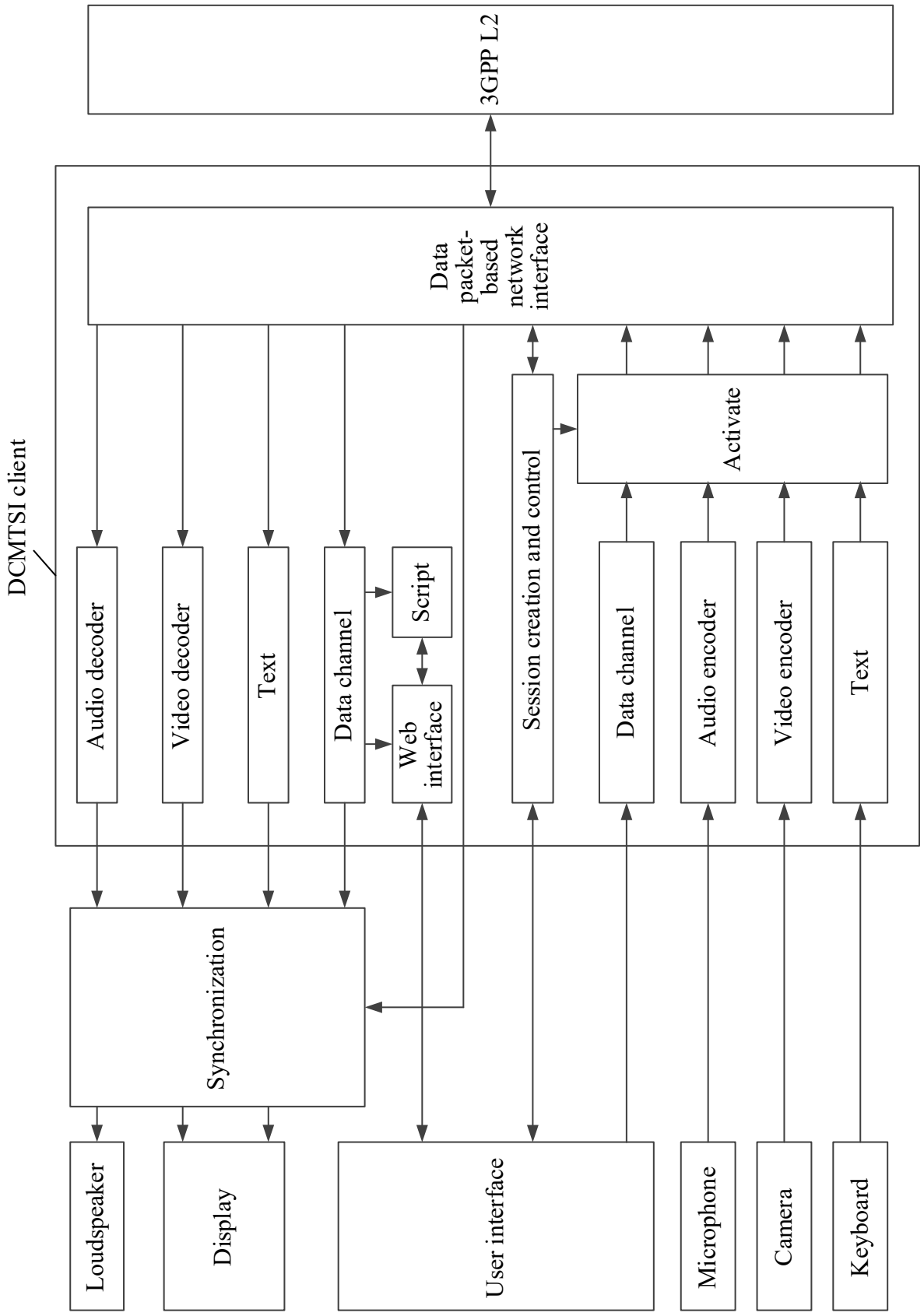

3GPP L2
DCMTSI client
Data packet-based network interface
Audio decoder
Video decoder
Text
Data channel
Script
Web interface
Session creation and control
Activate
Data channel
Audio encoder
Video encoder
Text
Synchronization
Loudspeaker
Display
User interface
Microphone
Camera
Keyboard

FIG. 3C

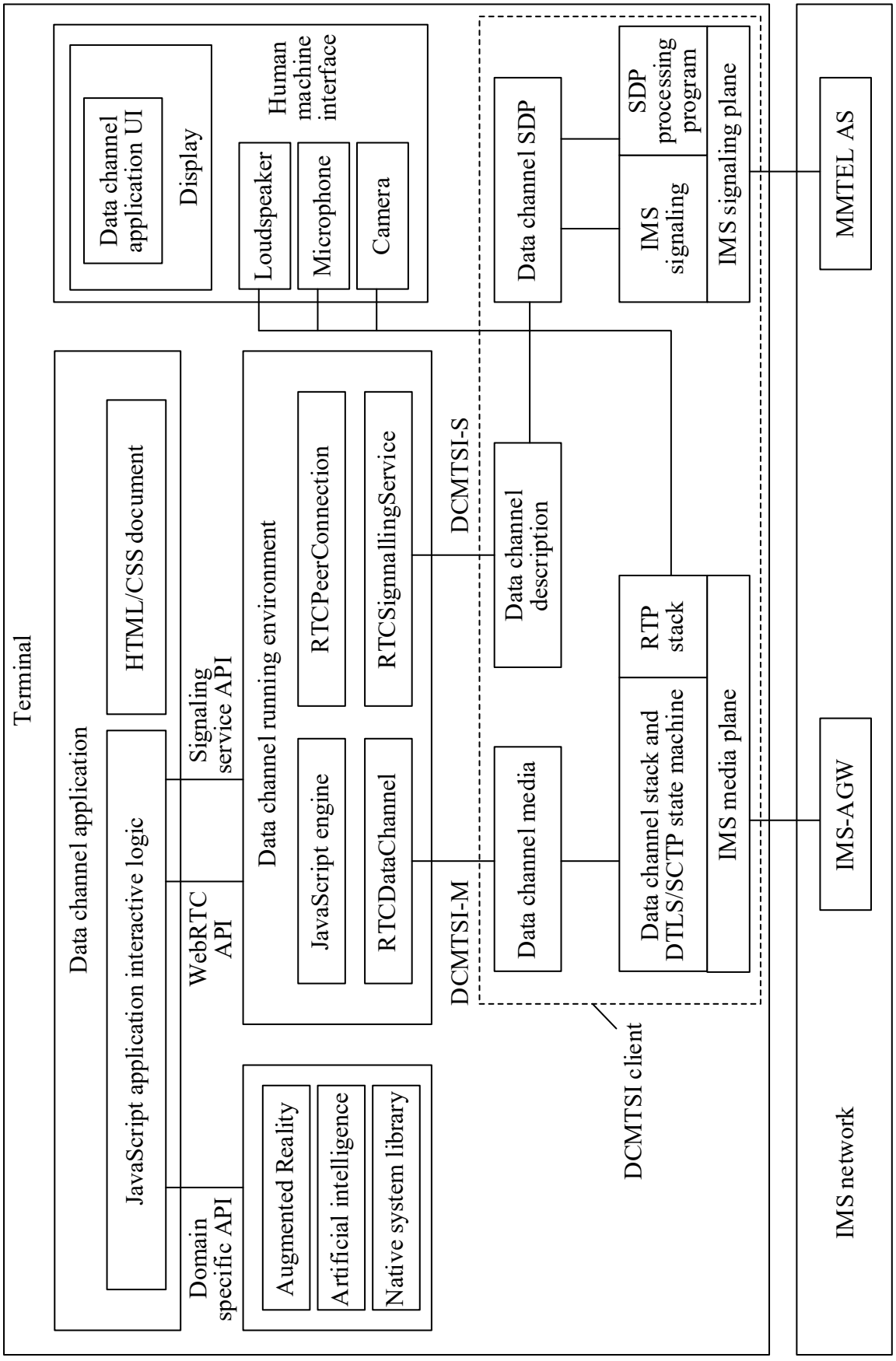

Terminal
Data channel application
JavaScript application interactive logic
HTML/CSS document
Domain specific API
WebRTC API
Signaling service API
Augmented Reality
Artificial intelligence
Native system library
Data channel running environment
JavaScript engine
RTCDataChannel
RTCPeerConnection
RTCSignnallingService
DCMTSI-S
DCMTSI-M
DCMTSI client
Data channel media
Data channel description
Data channel stack and DTLS/SCTP state machine
RTP stack
IMS media plane
Data channel SDP
IMS signaling
SDP processing program
IMS signaling plane
Human machine interface
Data channel application UI
Display
Loudspeaker
Microphone
Camera
IMS network
IMS-AGW
MMTEL AS

FIG. 3D

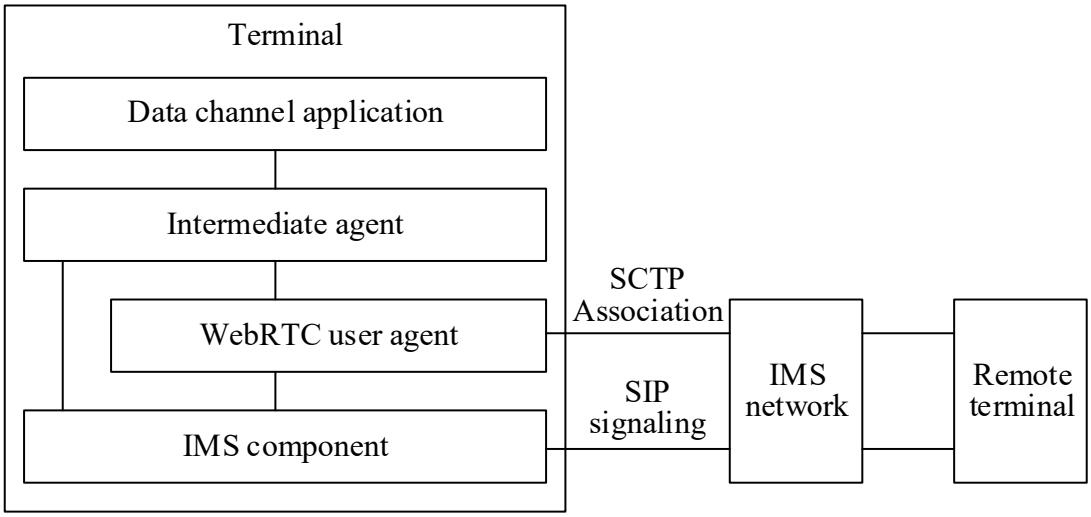


FIG. 3E

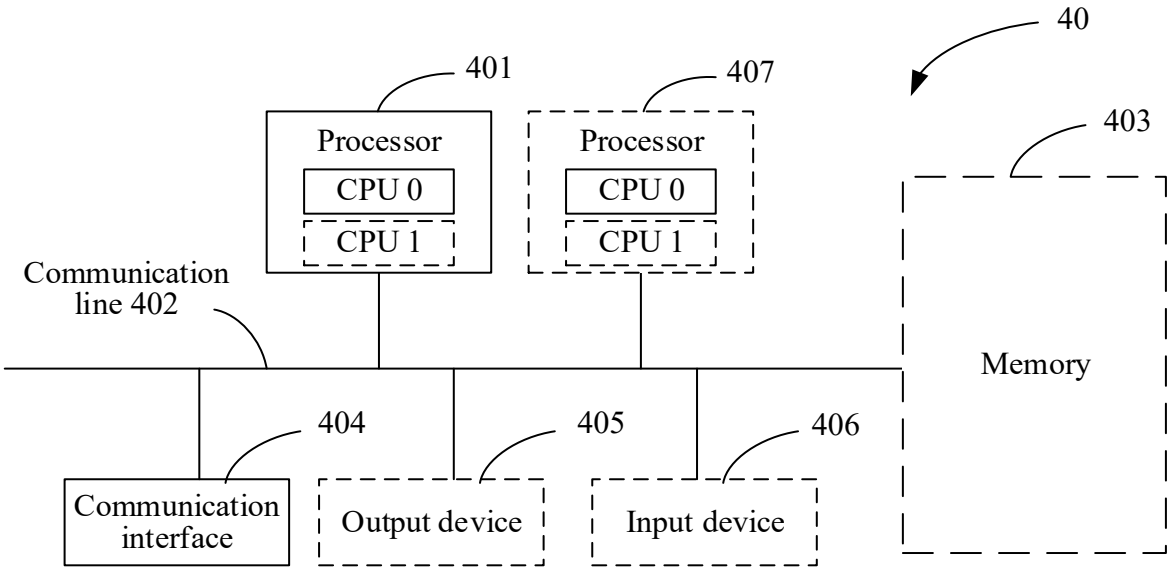


FIG. 4

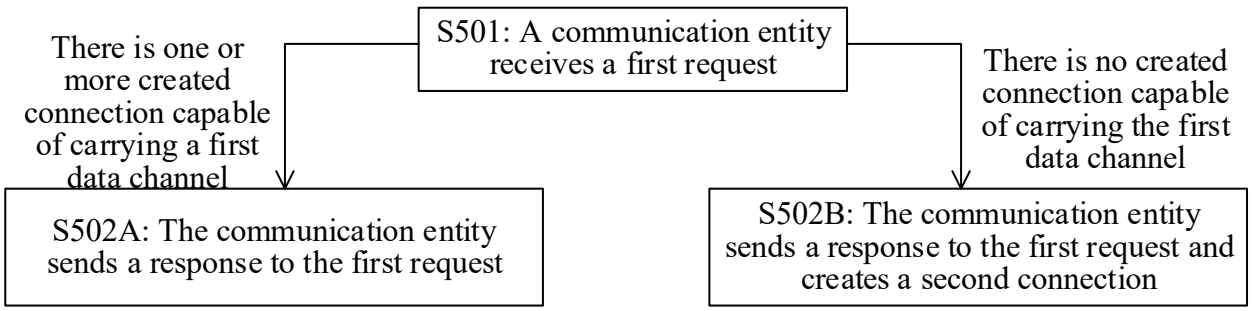


FIG. 5

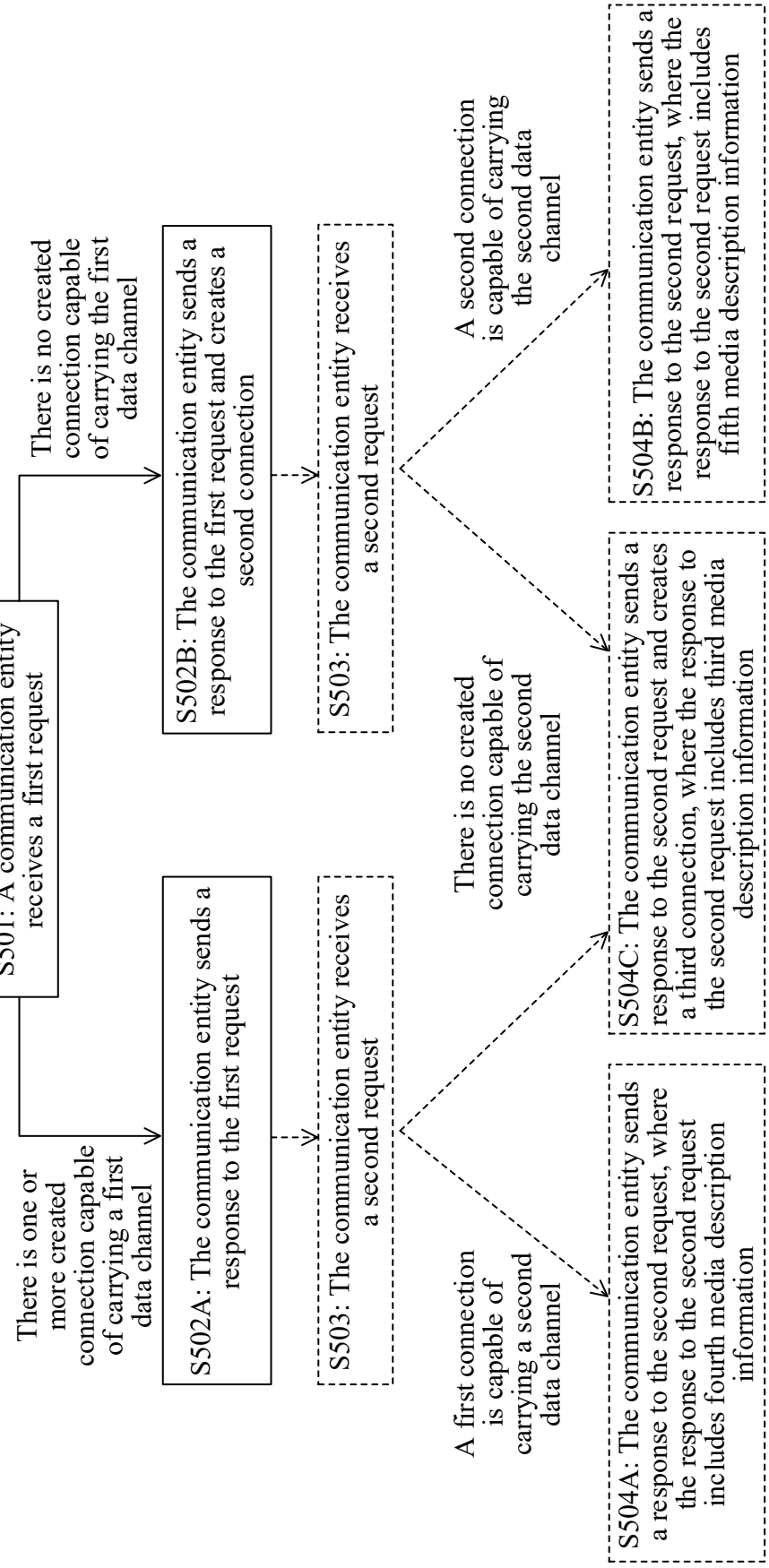

S501: A communication entity receives a first request
There is one or more created connection capable of carrying a first data channel
There is no created connection capable of carrying the first data channel
S502A: The communication entity sends a response to the first request
S502B: The communication entity sends a response to the first request and creates a second connection
S503: The communication entity receives a second request
S503: The communication entity receives a second request
A first connection is capable of carrying a second data channel
There is no created connection capable of carrying the second data channel
A second connection is capable of carrying the second data channel
S504A: The communication entity sends a response to the second request, where the response to the second request includes fourth media description information
S504C: The communication entity sends a response to the second request and creates a third connection, where the response to the second request includes third media description information
S504B: The communication entity sends a response to the second request, where the response to the second request includes fifth media description information

FIG. 6

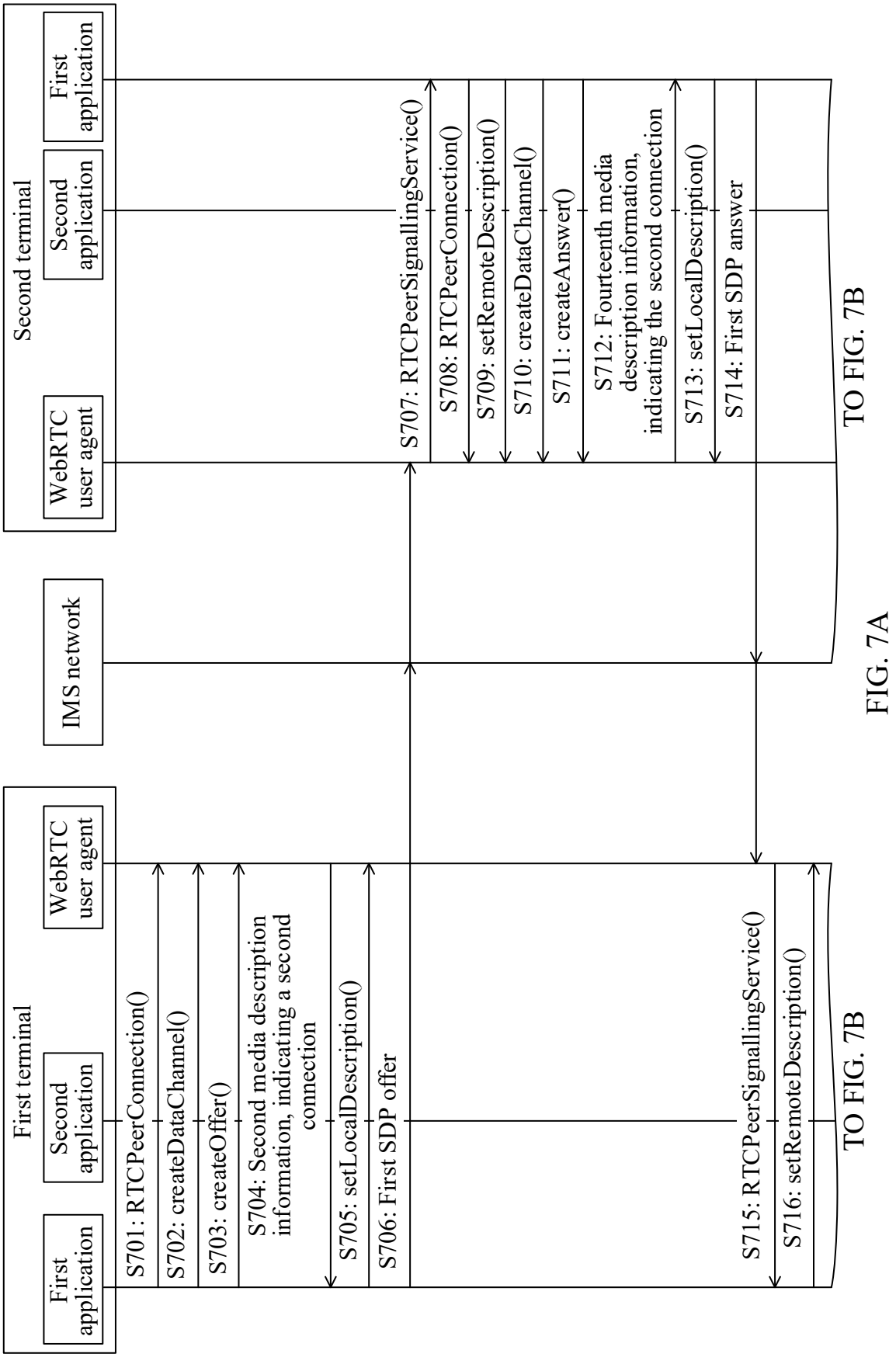

First terminal
First application
Second application
WebRTC user agent
IMS network
Second terminal
WebRTC user agent
Second application
First application
S701: RTCPeerConnection()
S702: createDataChannel()
S703: createOffer()
S704: Second media description information, indicating a second connection
S705: setLocalDescription()
S706: First SDP offer
S707: RTCPeerSignallingService()
S708: RTCPeerConnection()
S709: setRemoteDescription()
S710: createDataChannel()
S711: createAnswer()
S712: Fourteenth media description information, indicating the second connection
S713: setLocalDescription()
S714: First SDP answer
S715: RTCPeerSignallingService()
S716: setRemoteDescription()
TO FIG. 7B
TO FIG. 7B

FIG. 7A

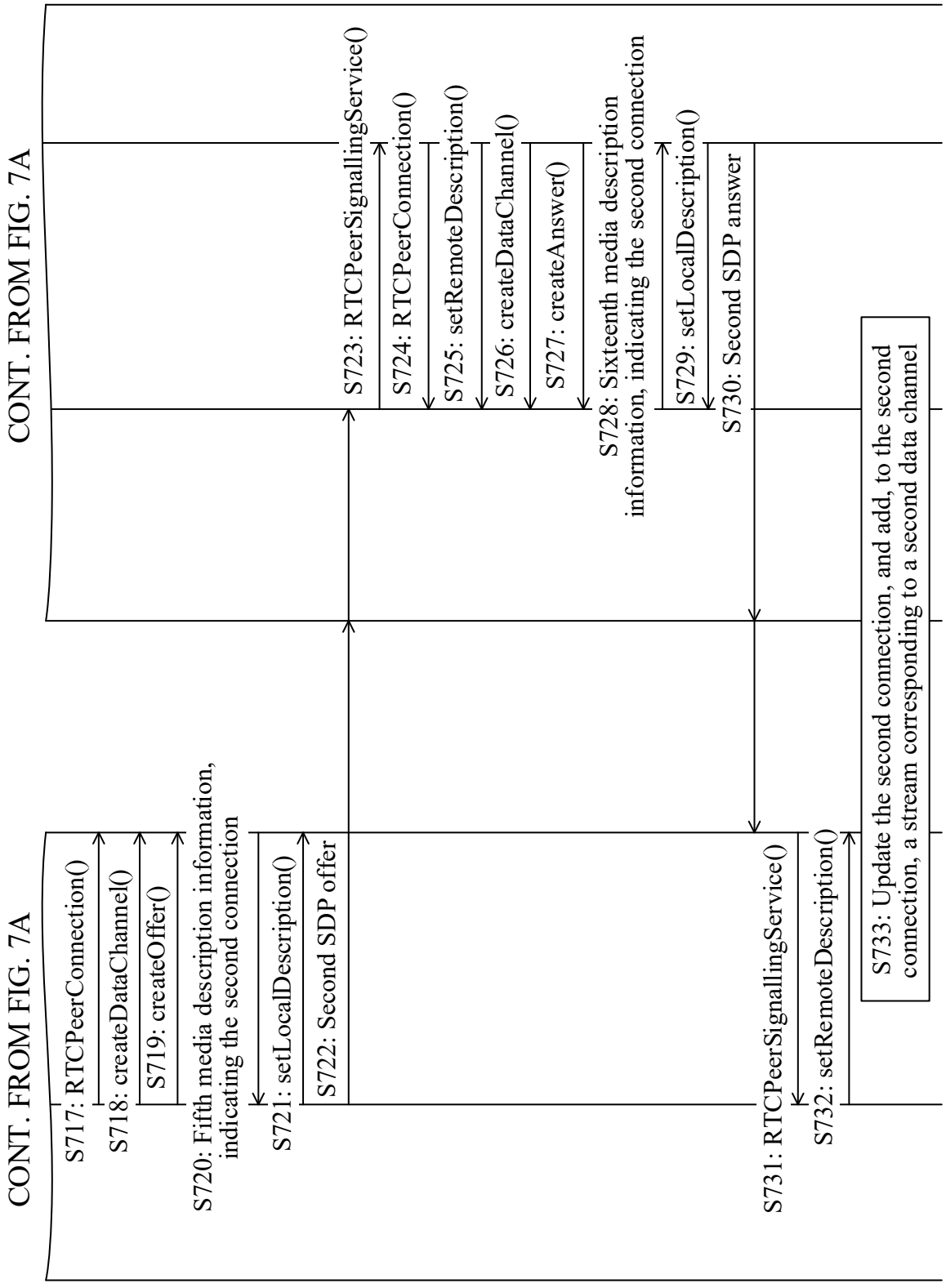

CONT. FROM FIG. 7A
S717: RTCPeerConnection()
S718: createDataChannel()
S719: createOffer()
S720: Fifth media description information, indicating the second connection
S721: setLocalDescription()
S722: Second SDP offer
S723: RTCPeerSignallingService()
S724: RTCPeerConnection()
S725: setRemoteDescription()
S726: createDataChannel()
S727: createAnswer()
S728: Sixteenth media description information, indicating the second connection
S729: setLocalDescription()
S730: Second SDP answer
S731: RTCPeerSignallingService()
S732: setRemoteDescription()
S733: Update the second connection, and add, to the second connection, a stream corresponding to a second data channel
CONT. FROM FIG. 7A

FIG. 7B

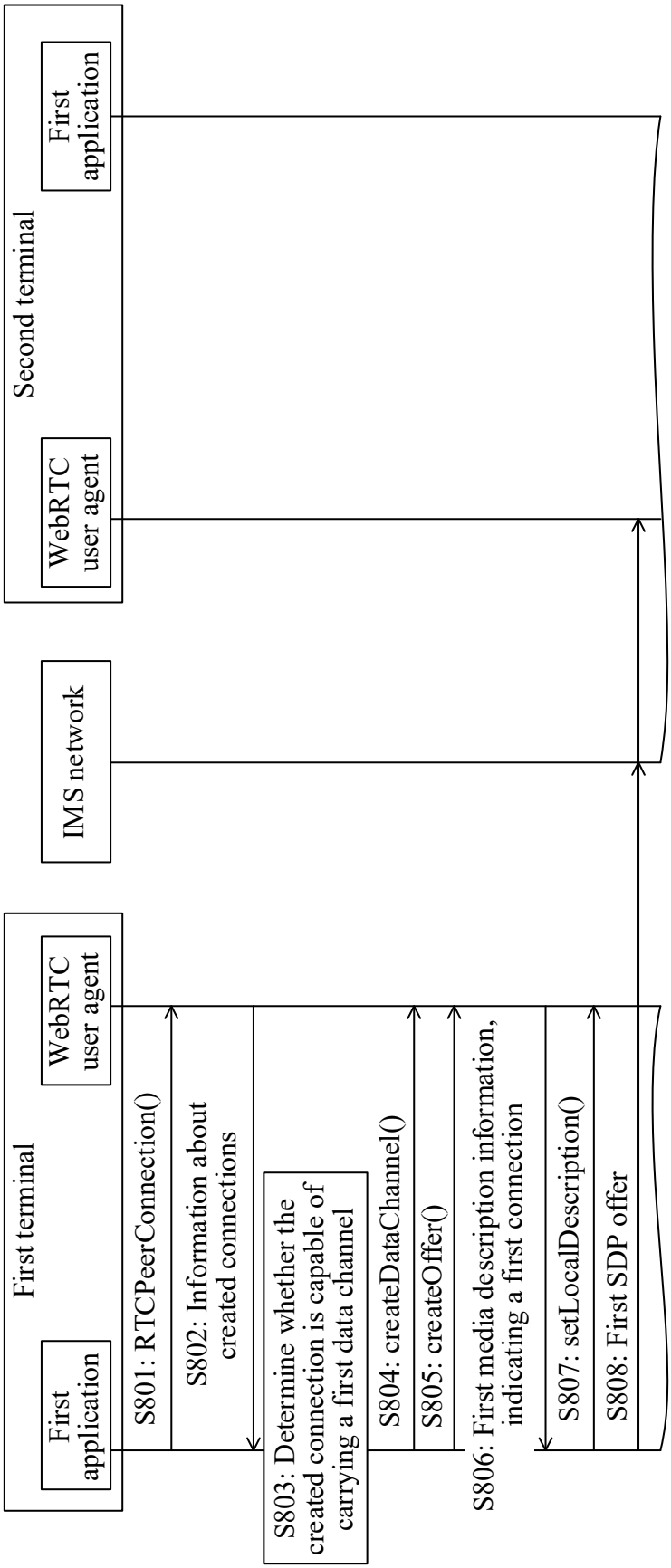

First terminal
First application
WebRTC user agent
IMS network
Second terminal
WebRTC user agent
First application
S801: RTCPeerConnection()
S802: Information about created connections
S803: Determine whether the created connection is capable of carrying a first data channel
S804: createDataChannel()
S805: createOffer()
S806: First media description information, indicating a first connection
S807: setLocalDescription()
S808: First SDP offer
TO FIG. 8B
TO FIG. 8B

FIG. 8A

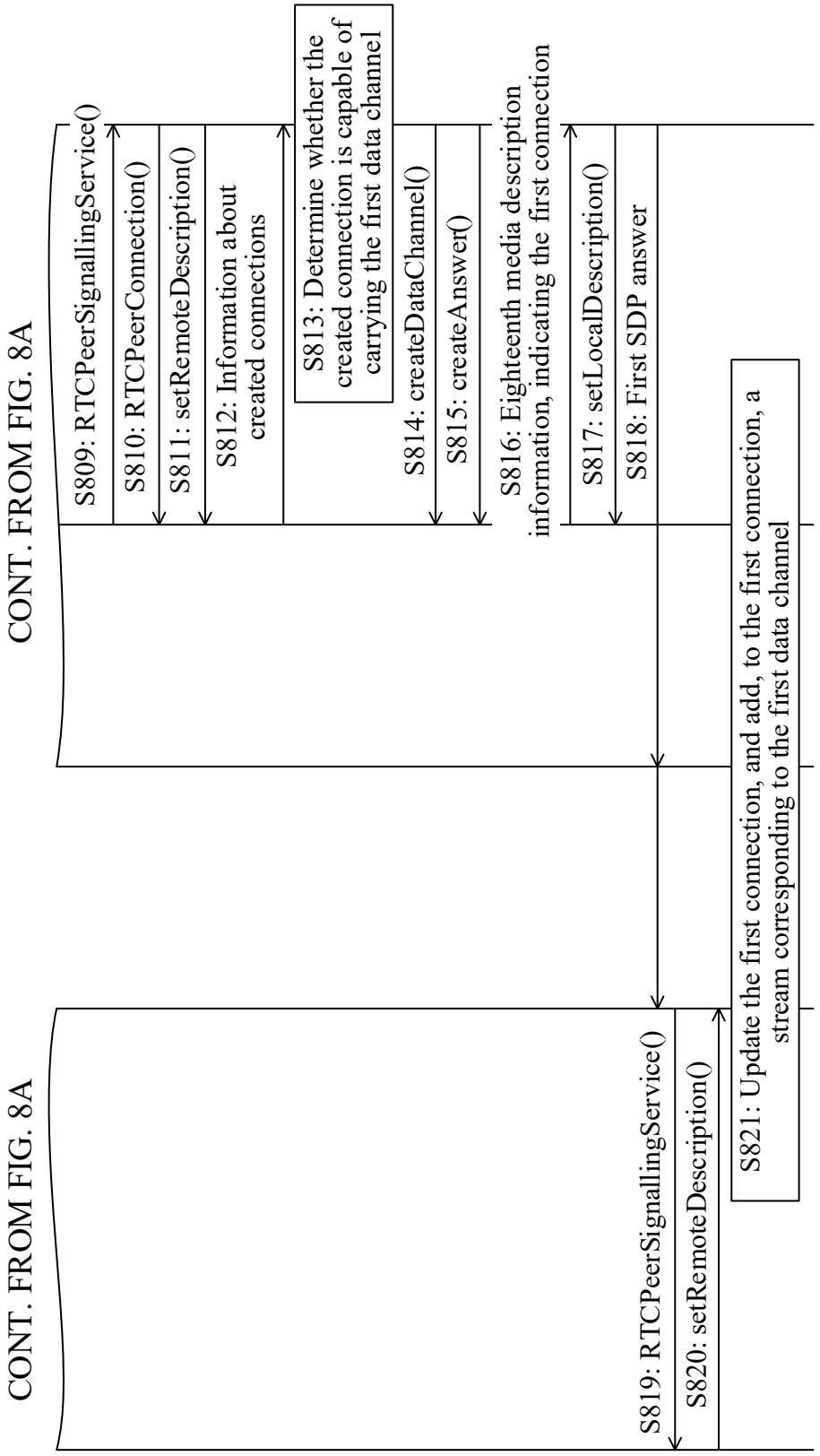

CONT. FROM FIG. 8A
S809: RTCPeerSignallingService()
S810: RTCPeerConnection()
S811: setRemoteDescription()
S812: Information about created connections
S813: Determine whether the created connection is capable of carrying the first data channel
S814: createDataChannel()
S815: createAnswer()
S816: Eighteenth media description information, indicating the first connection
S817: setLocalDescription()
S818: First SDP answer
CONT. FROM FIG. 8A
S819: RTCPeerSignallingService()
S820: setRemoteDescription()
S821: Update the first connection, and add, to the first connection, a stream corresponding to the first data channel

FIG. 8B

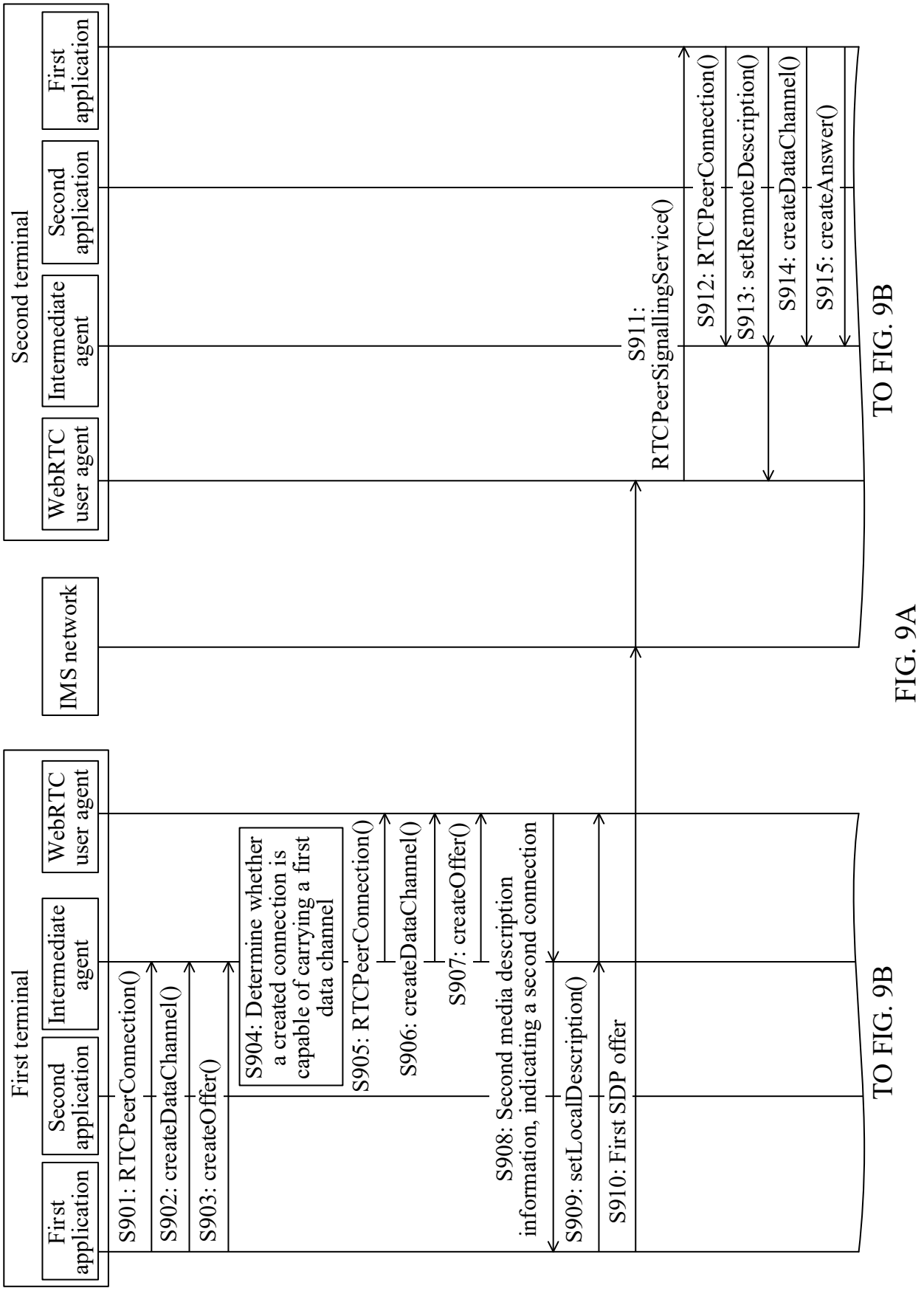

First terminal
First application
Second application
Intermediate agent
WebRTC user agent
IMS network
Second terminal
WebRTC user agent
Intermediate agent
Second application
First application
S901: RTCPeerConnection()
S902: createDataChannel()
S903: createOffer()
S904: Determine whether a created connection is capable of carrying a first data channel
S905: RTCPeerConnection()
S906: createDataChannel()
S907: createOffer()
S908: Second media description information, indicating a second connection
S909: setLocalDescription()
S910: First SDP offer
S911: RTCPeerSignallingService()
S912: RTCPeerConnection()
S913: setRemoteDescription()
S914: createDataChannel()
S915: createAnswer()
TO FIG. 9B
TO FIG. 9B

FIG. 9A

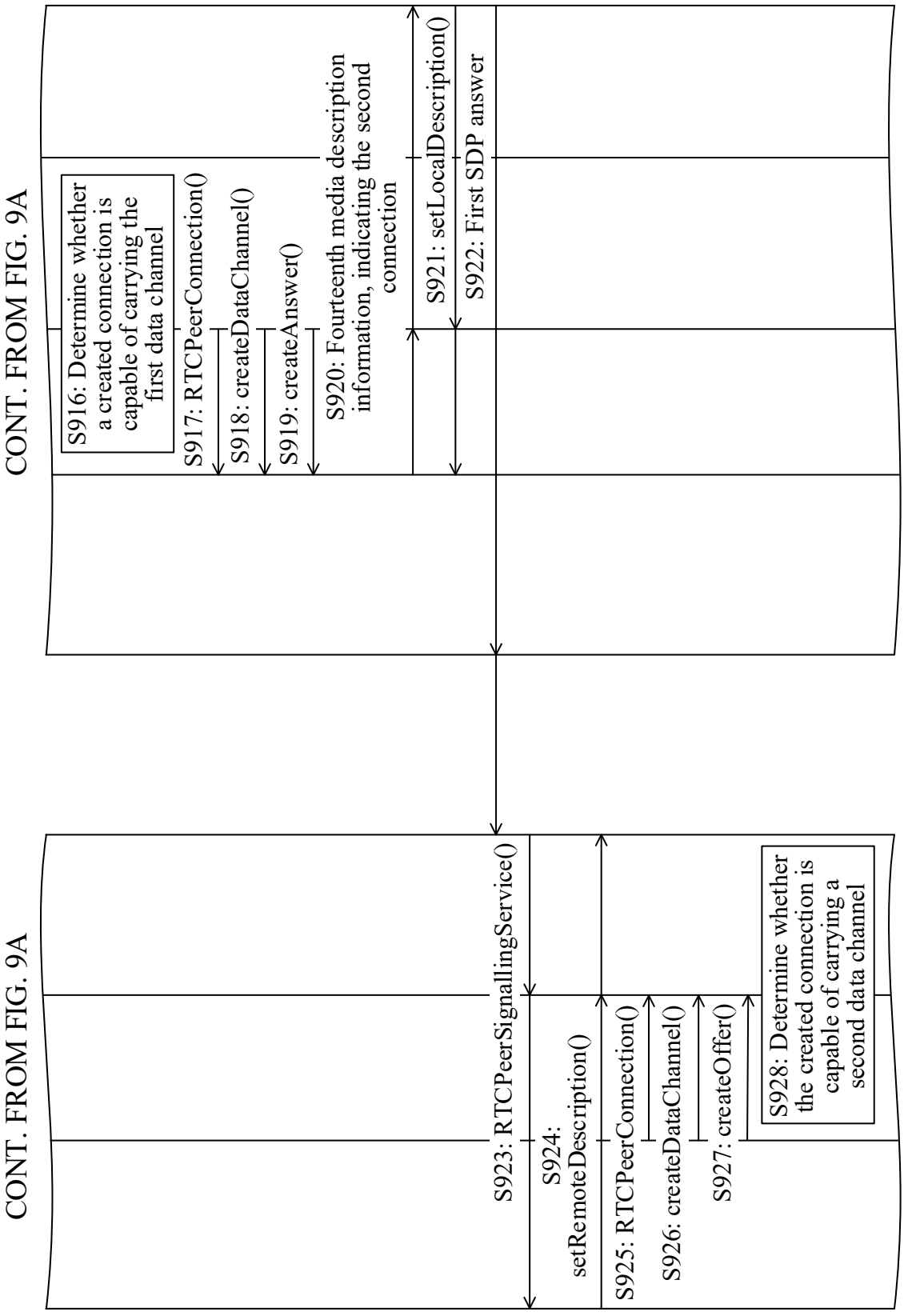

CONT. FROM FIG. 9A
S916: Determine whether a created connection is capable of carrying the first data channel
S917: RTCPeerConnection()
S918: createDataChannel()
S919: createAnswer()
S920: Fourteenth media description information, indicating the second connection
S921: setLocalDescription()
S922: First SDP answer
TO FIG. 9C
CONT. FROM FIG. 9A
S923: RTCPeerSignallingService()
S924: setRemoteDescription()
S925: RTCPeerConnection()
S926: createDataChannel()
S927: createOffer()
S928: Determine whether the created connection is capable of carrying a second data channel
TO FIG. 9C

FIG. 9B

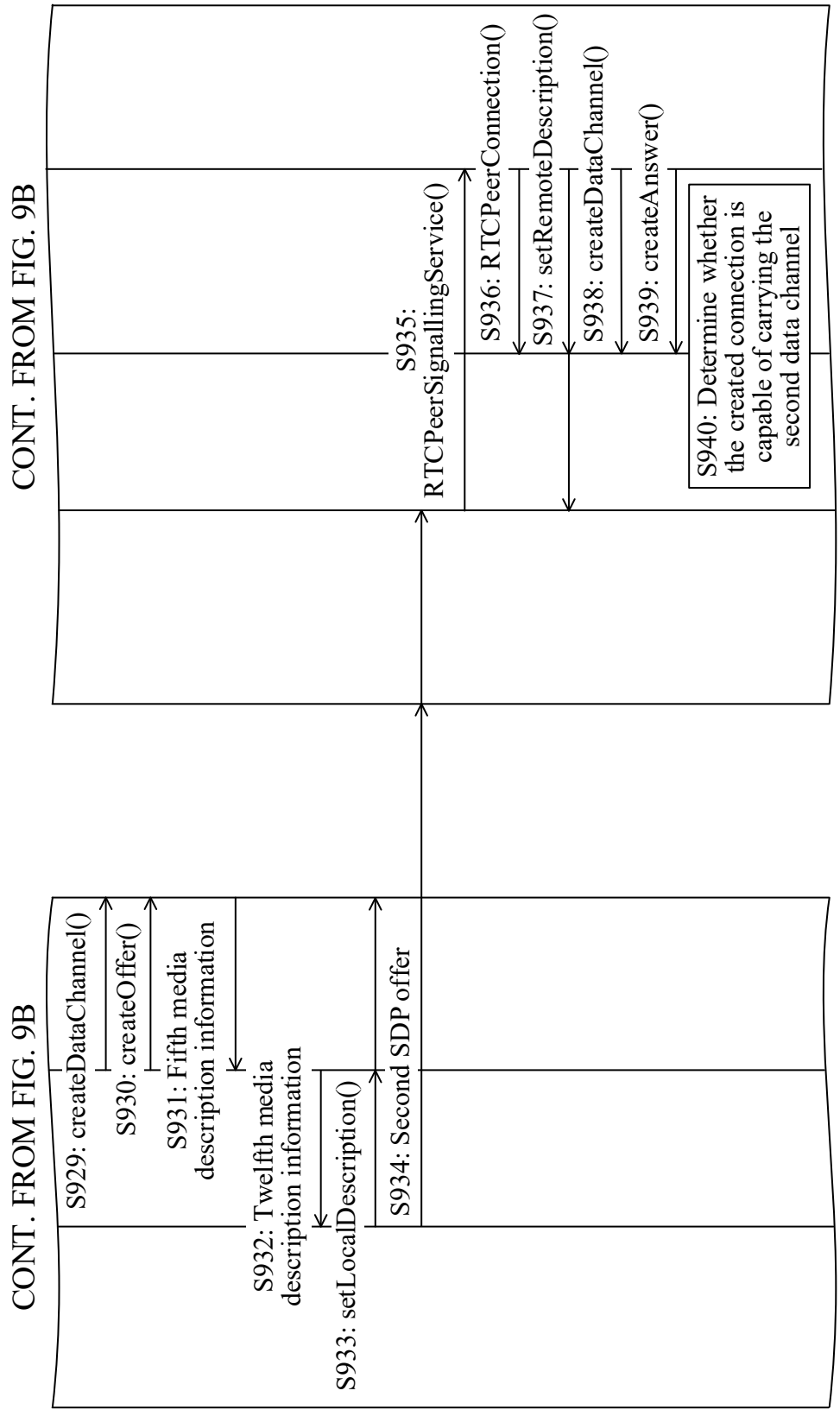

CONT. FROM FIG. 9B
S929: createDataChannel()
S930: createOffer()
S931: Fifth media description information
S932: Twelfth media description information
S933: setLocalDescription()
S934: Second SDP offer
TO FIG. 9D
CONT. FROM FIG. 9B
S935: RTCPeerSignallingService()
S936: RTCPeerConnection()
S937: setRemoteDescription()
S938: createDataChannel()
S939: createAnswer()
S940: Determine whether the created connection is capable of carrying the second data channel
TO FIG. 9D

FIG. 9C

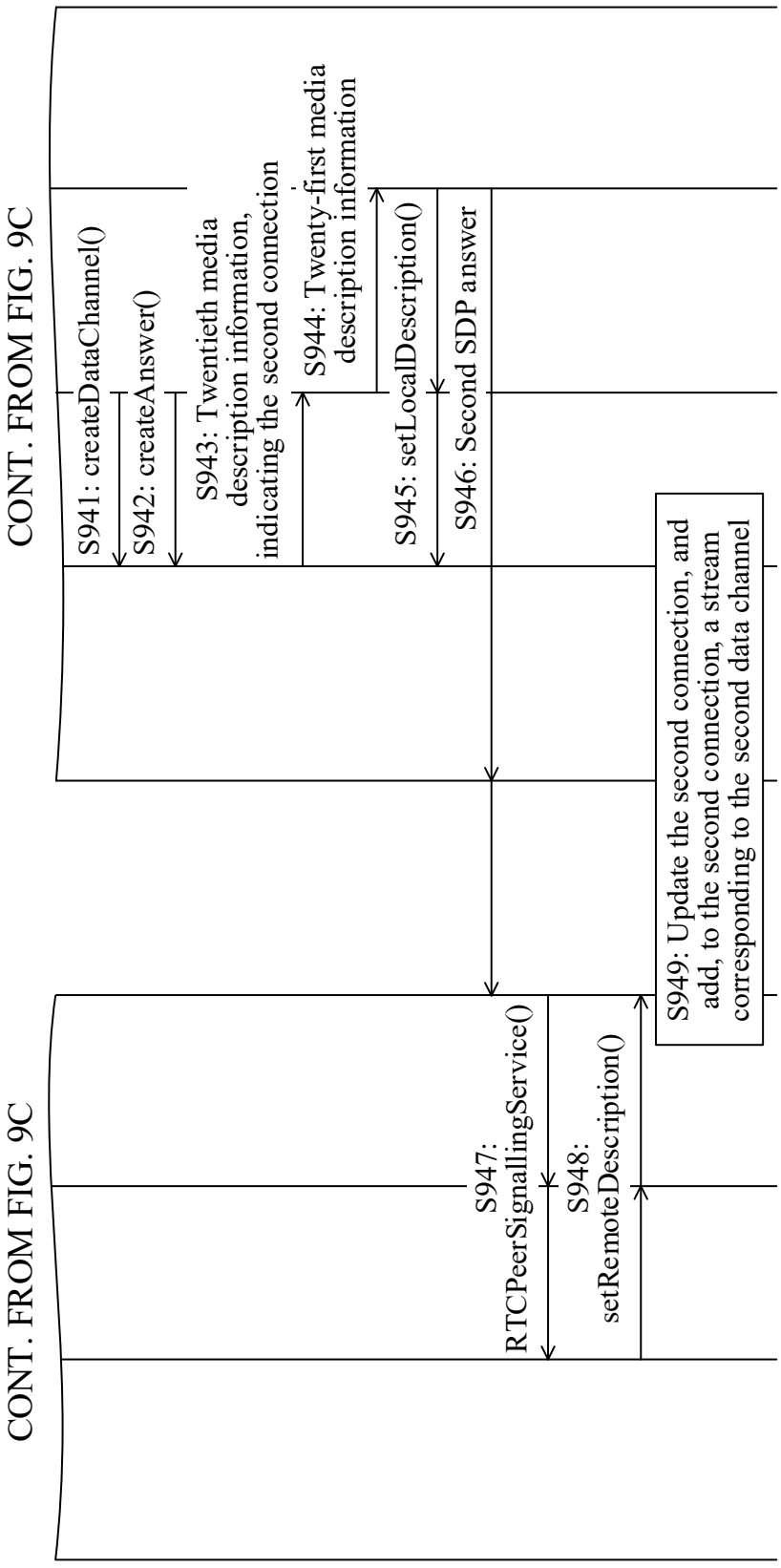

CONT. FROM FIG. 9C
S941: createDataChannel()
S942: createAnswer()
S943: Twentieth media description information, indicating the second connection
S944: Twenty-first media description information
S945: setLocalDescription()
S946: Second SDP answer
CONT. FROM FIG. 9C
S947: RTCPeerSignallingService()
S948: setRemoteDescription()
S949: Update the second connection, and add, to the second connection, a stream corresponding to the second data channel

FIG. 9D

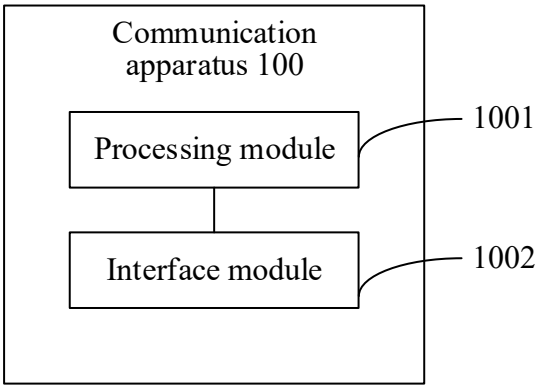

Communication apparatus 100
Processing module
1001
Interface module
1002

FIG. 10