**[API REVISED]**

**4) Should we provide a way to tell user whether the default adapter was removed. (Arthur)**

BluetoothAdapterEvent [1] would carry additional boolean isDefault to indicate whether the added/removed adapter is default adapter. See onadapteradded/onadapterremoved [2] for respective parameters.
[1] https://wiki.mozilla.org/B2G/Bluetooth/WebBluetooth-v2/BluetoothAdapterEvent
[2] https://wiki.mozilla.org/B2G/Bluetooth/WebBluetooth-v2/BluetoothManager#onadapteradded

**26) We should find a way to avoid to put address as a parameter of setPassKey. It's easy to make mistake for web developer since the address is just a DOMString. (Gene)**

BluetoothPairingEvent [3] would carry BluetoothDevice object and method setPasskey / setPairingConfirmation. See pairing event handlers [4] for respective parameters.
[3] https://wiki.mozilla.org/B2G/Bluetooth/WebBluetooth-v2/BluetoothPairingEvent
[4] https://wiki.mozilla.org/B2G/Bluetooth/WebBluetooth-v2/BluetoothAdapter#ondisplaypasskeyreq

**[API UNCHANGED]**

**======= Bluetooth Manager =======**

**2) Why we return a BT address in** onadapterremoved **but return a adapter in** onadapteradded **? (Arthur)**
**Should we return BT address for both APIs for consistency ? (Arthur)**
**Could we return BT adapter in** onadapterremoved **? (Gene)**
**Could we remain adapter instance for a while after it was removed ? (Gene)**
**We may need the info. of removed adapter, e.g. UX requirement. (Gene)**

onadapterremoved gets only address since the adapter is already removed and all its attributes become invalid when onadapterremoved is fired. We keep this behavior for two reasons: 1) to be consistent as other event handlers that are fired AFTER the event occurs and 2) what should its state be if we keep the adapter? It's not 'disabled' but we don't want to add a new state especially for this rare case. Even if there may be some UX requires removed adapter's info in the future, we prefer to keep current design till there is more specific application requirement.

**5) How about we design one unified API to replace** BluetoothManager.onadapteradded **and** BluetoothManager.onadapterremoved **? (Gene)**

We want the event handlers' names be clear to know their purpose. Two distinct onadapteradded and onadapterremoved are clearer than single onadapterchanged.

**6) How to get adapters ? "reaonly attribute" or "function call" ?**

If we make adapters as readonly attribute, to be consistent as BluetoothAdapter and BluetoothDevice, an onattributechanged event handler should exist to carry the new update adapter array. However we think application only care about the added/removed adapter instead of whole new array that needs manual comparison.

**======= Bluetooth Adapter =======**

**11) Should we put 'discovering' and 'discoverable' into** BluetoothAdapterState **? (Gene)**
**29) Does adapter need more state, such as 'pairing', 'connecting' and 'connected' ? (Arthur)**

We expect the states of BluetoothAdapterState are mutual exclusive, however 'discovering', 'discoverable', 'pairing', 'connecting', and 'connected' states are not as they all cover the 'enabled' state. For 'discovering' and 'discoverable', 4 additional states are required for their combination. Also for 'pairing', 'connecting', and 'connected' states we cannot ensure correct state transition for incoming pairing/connection. Therefore we keep them as attributes.

**13) Should we rebind handlers for adapter after it's been disabled ? (Evelyn)**

YES. To be consistent with attributes, the event handlers should also be reset as default value (nullptr). Application should rebind all handlers after adapter is re-enabled.

**15) When the adapter state is 'enabled', does it means we could get a correct address at that time? (Arthur)**

YES. To be consistent to that all attributes are reset to default values before state becomes 'disabled', we decide to ensure that all attributes are available after state becomes 'enabled.'

**20) If we put devicefound listener into req of startDiscovery(), we have to hold that req object in some place for unregister purpose. (Evelyn)**
**21) When does DOM req end? When the onsuccess is fired or when it exits its live cycle? (Eric)**

YES, gaia has to hold the reference to the DOM request. The DOM request exists until both gaia and gecko release references to it. So as long as gaia and gecko holds the DOM request's references, gaia can always receive the ondevicedfound event fired by gecko.

**Can application start discovery while adapter is discovering?**

No. It gets onerror. Applications have to stop current discovery first.