📖

Notification principles

🌏 Global notification principles

▼ Notification principles

All notifications across the Hopin platform must adhere to the following principles, even in their MVP phases of development.

- **[Supportive]:** Notifications should help users achieve their event goals. Event goals may vary depending on the user type and the different types of events.
- **[Seamless]:** Notifications should act seamlessly as part of the event experience, without interrupting the user's objectives.
- **[Configurable]:** Users should have the ability to tailor their event experience and configure notification settings according to their preferences.
- **[Relevant]:** Notifications should inform users about things they care about (unique/private) at the right time(context aware) and have the ability to adapt to the user’s state.

▼ Notification classification

Each notification can be grouped into three high level categories, which helps with ensuring all notifications are context aware and exceptions are evaluated accordingly. Here are examples of this classification logic:

1. **Wayfinding Notifications:**
    a. My Agenda

b. Guided Events

c. Meetings

d. Breakout Rooms

2. **Chat Notifications**
   a. Mentions
   b. Threaded Replies
   c. Likes
   d. DMs

3. **Occurrence Based / Organizer Driven Notifications**
   a. General Announcements
   b. New Polls
   c. Submit Q&A
   d. Your Q&A has been answered

▼ Notification types & attributes

▼ **Notification types**

1. **(Only on native Mobile) Push:** Push notifications are clickable pop-up alerts that appear on users' mobile devices. Subscribers can be anywhere on their phone and still receive these alerts as long as they're online. Push notifications are proven to increase repeat visits, user retention, and re-engagement.

2. **In-App:** In-app notifications are built-in notifications that can present themselves without asking for permission on the Hopin platform. The difference between push notifications is that they're confined to the app itself.

   a. **<u>In-App Previews:</u>** Previews slide in from the right hand side of the screen and stay visible for 15 seconds.

   b. **<u>In-App Modal:</u>** This is the most disruptive type of notification on Web and is usually only used for critical information to be shared with attendees that require their immediate attention. Modals surface at the center of the screen and must be dismissed by the user directly.

c. **Counter Badge Alert + Stored in NC Tray:** A red oval containing a white number on top of the icons to indicate when new noncritical information is available. With each new alert, the information also gets stored in the NC so that users can access it at a later time.

▼ Notification attributes

Attributes are extensions that can be added to notification types that determine how they behave. Each notification can have one or multiple attributes at the same time.

1. **[Sound]:** Sound is tied with the in-app preview notification type (meaning all preview notifications will also have a sound component). Sounds are useful when users are multi-tasking or distracted and can bring them back to the event.
2. **[CTA]:** Call to actions take users to the place they need to be in order to make sense of incoming stimuli and easily take the appropriate action with the single click of a button.
3. **[Overlay]**: Overlays are tied with the in-app modal notification type (meaning all modal notifications will also have an overlay component). Overlays are the greying/blurring out of the background of the screen to bring the notification to the center of the user’s view.
4. **[Redirects]**: A redirect can take a form of a modal or sidebar and usually the user is being redirected to where he needs to be at the right time.

▼ Annoyance aversion & configuration control

Guardrails will be enforced to prevent annoyance. Sound and Preview alerts will be muted after a predefined threshold (15 seconds) is reached (meaning there will only be one preview/sound for any number of incoming notifications within a 15 second timeframe. We will also use the context map to ensure we only show notifications that are relevant to users which will reduce unnecessary noise. Users should have the ability to control notification settings (Designs here). Users should have the ability to control the types of notifications they receive and how they receive them eg. ability to turn off certain notification types, ability to mute/unmute sound alerts etc.

- When a user turns notifications off then turns them back on; the counterbadge alerts will only start counting from that moment onward (IOW, no previously received notifications will be re-counted)
- Any changes made to the settings are at the event-level and will revert back to the default settings when the same user attends a different event. This behavior is proposed based on keeping consistency across Hopin (low rates of account-level activity on Hopin) and aligned with our GTM Strategy.
- We are exploring whether to allow organizers to change defaults based on their objectives (eg. absorb event content, networking etc.) but the thinking is that users will have ultimate control and can override any organizer defined settings (with the exception of GA, which users will not have the ability to turn off).

▼ Accessibility requirements

- **Aim for inclusive experience:** Ensure to provide the same experience for users who have problems using devices. Like when the notification component has buttons or links, everyone should be able to interact with them
- **Store important messages in the notification center:** Make sure to store all important notifications in the notification center in case when users miss reading, hearing or seeing the notifications
- **Be considerate when to move focus:** When users are using interactive UIs (i.e. text inputs), focus should not move to the notification, but still be informed.
- **Give users enough time to read messages:** A good length of time to keep messages up is 5 seconds plus 1 extra second for every 120 words, rounding up. WCAG 2.1 success criterion suggests 15 seconds; 15 seconds to hit 'any switch' is sufficient for almost all users including those with spasticity. When the notification is not stored in the notification center, the user should be allowed to extend the time limit at least ten times.
- **Don’t use notifications that dismiss on a timer for critical or emergency messages:** Some users with disabilities need more time to read or interact with messages and timed toasts may not provide sufficient time.

👨‍💻 Attendee principles

💻 Web principles (Default)

▼ Notification types & attributes for web

▼ **Notification types**

1. **In-App:** In-app notifications are built-in notifications that can present themselves without asking for permission on the Hopin platform. The difference between push notifications is that they're confined to the app itself.

   a. **In-App Previews:** Previews slide in from the right hand side of the screen and stay visible for 15 seconds.

   b. **In-App Modal:** This is the most disruptive type of notification on Web and is usually only used for critical information to be shared with attendees that require their immediate attention. Modals surface at the center of the screen and must be dismissed by the user directly.

   c. **Counter Badge Alert + Stored in NC Tray:** A red oval containing a white number on top of the icons to indicate when new noncritical information is available. With each new alert, the information also gets stored in the NC so that users can access it at a later time.

▼ **Notification attributes**

Attributes are extensions that can be added to notification types that determine how they behave. Each notification can have one or multiple attributes at the same time.

1. **[Sound]:** Sound is tied with the in-app preview notification type (meaning all preview notifications will also have a sound component). Sounds are useful when users are multi-tasking or distracted and can bring them back to the event.
2. **[CTA]:** Call to actions take users to the place they need to be in order to make sense of incoming stimuli and easily take the appropriate action with the single click of a button.

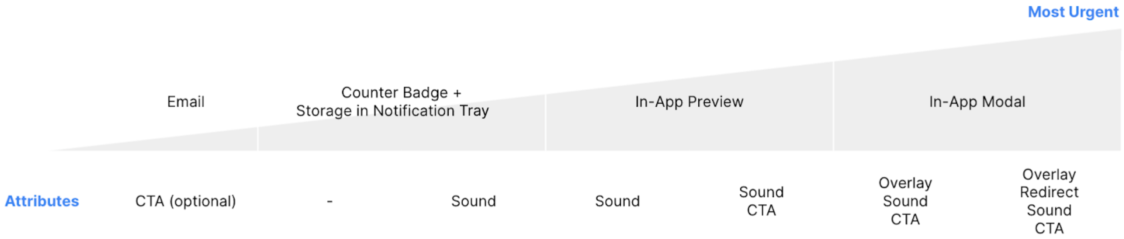
3. **[Overlay]**: Overlays are tied with the in-app modal notification type (meaning all modal notifications will also have an overlay component). Overlays are the greying/blurring out of the background of the screen to bring the notification to the center of the user's view.
4. **[Redirects]**: A redirect can take a form of a modal or sidebar and usually the user is being redirected to where he needs to be at the right time.

▼ Ranking & behaviour Logic

Urgency and importance are being measured based on the below in this order:

▼ **Type**

An assessment on the urgency of the notification should be the initial determination of how that notification should behave.



- **Visibility & Attributes: Detailed matrix can be found here [link]**
- **Context Aware Rules: Notifications must follow predefined rules and surface only when relevant to the user type, depending on where they are in the workflow.**

▼ **Context**

- **[Side bar is minimized]** Sidebar is minimized when attendees feel distracted and want to focus on the content of the event. However the use case here is for filtering out the general events that are not specific to the user. Notifications should still appear when the user minimizes the sidebar because they are the filtered view of alerts that are unique to the users that require their attention/them to take a specific action.
- **[Notifications are triggered based on the user type]** attendee vs speaker etc.

    - If a user is speaking or presenting, they should not receive preview and sound notifications

- **[Organiser driven]** - We show these types of notifications (polls/Q&As/Apps etc) only when the user is in the designated “space” (sessions/stage/expo)
- **[When in session, stage or expo]**
    - DO NOT SHOW GE & Agenda notifications if you’re in the right space
    - USE REDIRECT for organiser driven notifications ONLY for the space that I’m in (current poll, Q&A or app in my current stage, session or expo)
    - ADD event level organiser driven notifications at all times to the NOTIFICATIONS TRAY
- **[When in sidebar “Agenda” tab]**
    - If “Started broadcasting on Stage” is added to “my agenda” the “Started broadcasting on stage” notification should not appear
    - If in the Agenda tab, NO NOT SHOW preview or sound for agenda segments except meeting reminders
- **[Modal, nav dropdown, conf. dialog or DM is open]**
    - **Whenever you have a modal or a dropdown open, then preview IS SHOWN behind the dropdown or modal and all new notifications are being delayed/paused until user has closed the dropdown/modal.**
    - Whenever you are in DM or messaging in the sidebar then previews DO NOT work
    - Whenever you are in DM messaging the same person, then NO NOTIFICATIONS ARE SHOWN
    - Whenever you have the NC tray open, then NO PREVIEWS ARE SHOWN
    - Whenever you click on an item in the notification tray or preview, you’re taken to where the notification was originating from

- If a notification is taking me to a different from my current space and I’m participating in a session, stage or 1:1/group networking, then a confirm dialog must be shown

▼ **Trumping Behaviour**

User receives multiple different notifications at the same time, which means they need to be ordered by importance and urgency in mind. A specific ordering has been assigned to each notification type which can be found here.

1. Stand alone critical notifications should be served one by one in that order:
    a. System notifications
    b. GA
    c. GE
    d. Doors open
    e. Breakout room created
2. All grouped notifications can come at the same time, but the ones on top would appear with greater importance
3. Similar notification previews are grouped with the most recent one surfacing on top

📱 Mobile native principles

▼ **Mobile specific considerations & guidelines**

▼ Consent needed

User must give consent for push notifications (Native banners, sound and icon badges) on iOS

> 🍏 Detailed consent guidelines for iOS on here

▼ Native mobile guidelines for notifications

🍏 iOS Notification guidelines

🤖 Android guidelines | Material design

▼ Actionable notifications need to be declared

Although it's not required, every notification should open an appropriate app activity when tapped. In addition to this default notification action, you can add action buttons that complete an app-related task from the notification (often without opening an activity)

🤖 Android guidelines

🍏 iOS guidelines

▼ Notification settings need to be native

All notification controls and settings need to live in the app (channel) settings area

▼ Trumping behaviour on mobile

Trumping behaviour is additionally shaped by Interruption & importance levels both of which are outlined below

▼ Notification types & attributes for mobile

▼ **Notification types**

1. **Mobile native push: *(Push notifications replace previews on mobile)***
   Push notifications are clickable pop-up alerts that appear on users' mobile devices. Subscribers can be anywhere on their phone and still receive these alerts as long as they're online. Push notifications are proven to increase repeat visits, user retention, and re-engagement. On iOS they're

called ***banners*** while on Android they're referred to as ***heads-up notifications***

2. **In-App:** In-app notifications are built-in notifications that can present themselves without asking for permission on the Hopin platform. The difference between push notifications is that they're confined to the app itself.

    a. **<u>In-App Modal:</u>** This is the most disruptive type of notification on Web and is usually only used for critical information to be shared with attendees that require their immediate attention. Modals surface at the center of the screen and must be dismissed by the user directly.

    b. **<u>Counter Badge Alert + Stored in NC Tray:</u>** A red oval containing a white number on top of the icons to indicate when new noncritical information is available. With each new alert, the information also gets stored in the NC so that users can access it at a later time.

▼ **Notification attributes**

Attributes are extensions that can be added to notification types that determine how they behave. Each notification can have one or multiple attributes at the same time.

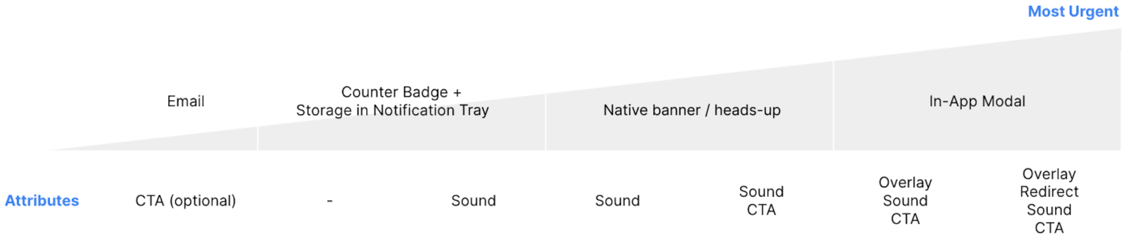
1. **[Sound]:** Sound is tied with the **push notification type** and it's handled natively (meaning all push notifications have a sound component as long as a user has given consent). Sounds are useful when users are multi-tasking or distracted and can bring them back to the event.
2. **[CTA]:** Call to actions take users to the place they need to be in order to make sense of incoming stimuli and easily take the appropriate action with the single click of a button.
3. **[Overlay]**: Overlays are tied with the in-app modal notification type (meaning all modal notifications will also have an overlay component). Overlays are the greying/blurring out of the background of the screen to bring the notification to the center of the user's view.
4. **[Redirects]**: A redirect can take a form of a modal or sidebar and usually the user is being redirected to where he needs to be at the right time.

▼ Ranking & behaviour logic

Urgency and importance are being measured based on the below in this order:

▼ Type

An assessment on the urgency of the notification should be the initial determination of how that notification should behave.



Most Urgent

	Email	Counter Badge + Storage in Notification Tray		Native banner / heads-up		In-App Modal	
Attributes	CTA (optional)	-	Sound	Sound	Sound CTA	Overlay Sound CTA	Overlay Redirect Sound CTA

- **Visibility & Attributes: Detailed matrix can be found here [link]**
- **Context Aware Rules: Notifications must follow predefined rules and surface only when relevant to the user type, depending on where they are in the workflow.**

▼ Context

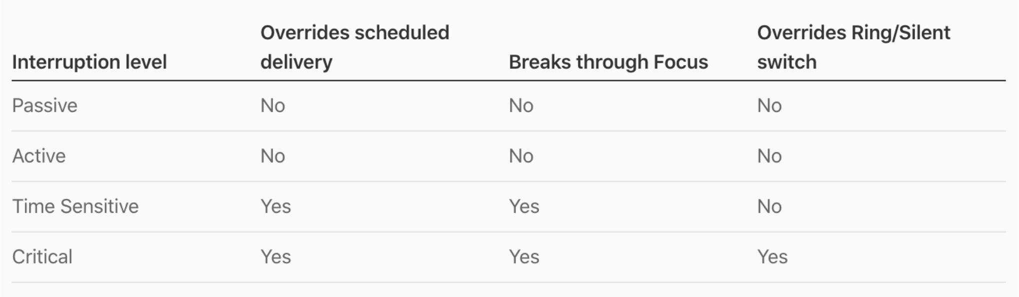
- **[Side bar is minimized]** Sidebar is minimized when attendees feel distracted and want to focus on the content of the event. However the use case here is for filtering out the general events that are not specific to the user. Notifications should still appear when the user minimizes the sidebar because they are the filtered view of alerts that are unique to the users that require their attention/them to take a specific action.
- **[Notifications are triggered based on the user type]** attendee vs speaker etc.
  - If a user is speaking or presenting, they should not receive preview and sound notifications
- **[Organiser driven]** - We show these types of notifications (polls/Q&As/Apps etc) only when the user is in the designated “space” (sessions/stage/expo)
- **[When in session, stage or expo]**
  - DO NOT SHOW GE & Agenda notifications if you’re in the right space

  - USE REDIRECT for organiser driven notifications ONLY for the space that I’m in (current poll, Q&A or app in my current stage, session or expo)
  - ADD event level organiser driven notifications at all times to the NOTIFICATIONS TRAY
- **[When in sidebar “Agenda” tab]**
  - If “Started broadcasting on Stage” is added to “my agenda” the “Started broadcasting on stage” notification should not appear
  - If in the Agenda tab, NO NOT SHOW preview or sound for agenda segments except meeting reminders
- **[Modal, nav dropdown, conf. dialog or DM is open]**
  - **Whenever you have a modal or a dropdown open, then preview IS SHOWN behind the dropdown or modal and all new notifications are being delayed/paused until user has closed the dropdown/modal.**
  - Whenever you are in DM or messaging in the sidebar then previews DO NOT work
  - Whenever you are in DM messaging the same person, then NO NOTIFICATIONS ARE SHOWN
  - Whenever you have the NC tray open, then NO PREVIEWS ARE SHOWN
  - Whenever you click on an item in the notification tray or preview, you’re taken to where the notification was originating from
  - If a notification is taking me to a different from my current space and I’m participating in a session, stage or 1:1/group networking, then a confirm dialog must be shown

▼ **Trumping Behaviour**

User receives multiple different notifications at the same time, which means they need to be ordered by importance and urgency in mind. A specific ordering has been assigned to each notification type which can be found here.

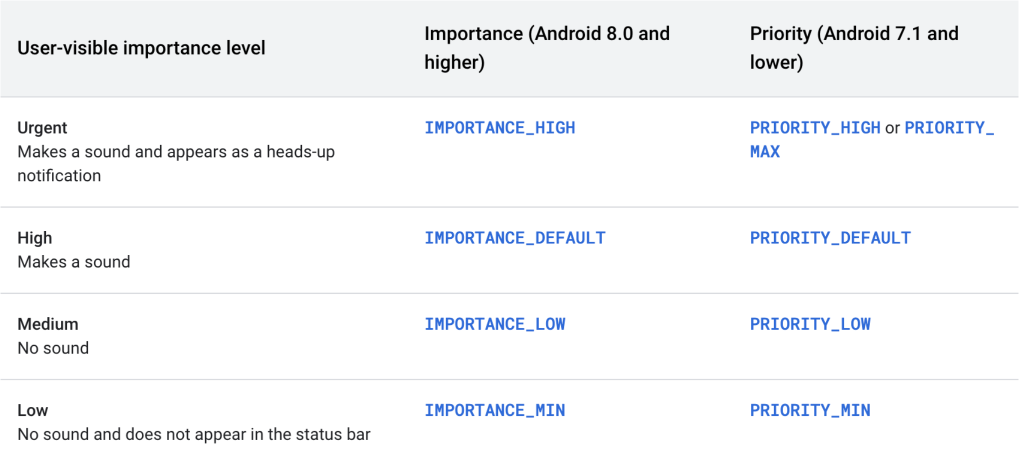
1. **Interruption/importance levels**
    a. Detailed behaviour breakdown on here



Interruption level	Overrides scheduled delivery	Breaks through Focus	Overrides Ring/Silent switch
Passive	No	No	No
Active	No	No	No
Time Sensitive	Yes	Yes	No
Critical	Yes	Yes	Yes

(🍏 iOS guidelines)

    b. Detailed behaviour breakdown on here



User-visible importance level	Importance (Android 8.0 and higher)	Priority (Android 7.1 and lower)
Urgent Makes a sound and appears as a heads-up notification	`IMPORTANCE_HIGH`	`PRIORITY_HIGH` or `PRIORITY_MAX`
High Makes a sound	`IMPORTANCE_DEFAULT`	`PRIORITY_DEFAULT`
Medium No sound	`IMPORTANCE_LOW`	`PRIORITY_LOW`
Low No sound and does not appear in the status bar	`IMPORTANCE_MIN`	`PRIORITY_MIN`

(🤖 Android guidelines)

2. Stand alone critical notifications should be served one by one in that order:
    a. System notifications
    b. GA
    c. GE
    d. Doors open
    e. Breakout room created

3. All grouped notifications can come at the same time, but the ones on top would appear with greater importance
4. Similar notification previews are grouped with the most recent one surfacing on top

💌 Email principles

▼ Notification type and attributes

> 📕 Detailed mail matrix breakdown can be found here

Email: Alerts that do not require users’ immediate attention but that can be accessed throughout the entire event lifecycle. Email alerts are also important for hybrid events where attendees may be on the move.

▼ **Notification attributes**

Attributes are extensions that can be added to notification types that determine how they behave. Each notification can have one or multiple attributes at the same time.

1. **[CTA]:** Call to actions take users to the place they need to be in order to make sense of incoming stimuli and easily take the appropriate action with the single click of a button.