

# Обработване на събития *(Event Handling)*

# СЪБИТИЯ (или *Event-и*)

- света на Flash се върти около събитията
- механизъмът за обработка на събития е базиран на стандарт (*DOM event handling standard*)
- за да може компонентът ни да излъчва събития (*dispatch events*) и да позволява на други да добавят и махат слушатели (*listeners*) за тези събития, той трябва да имплементира *IEventDispatcher* интерфейса
- най-често използваме готовата имплементация на класа *EventDispatcher*
- всички слушатели получават *Event*-обект като аргумент

# flash.events.IEventDispatcher

```
addEventListener(type:String, listener:Function,  
    useCapture:Boolean=false, priority:int=0,  
    useWeakReference:Boolean=false):void;
```

```
dispatchEvent(event:Event):Boolean;
```

```
hasEventListener(type:String):Boolean;
```

```
removeEventListener(type:String, listener:Function,  
    useCapture:Boolean=false):void;
```

```
willTrigger(type:String):Boolean;
```

# flash.events.Event

```
bubbles:Boolean;
```

```
cancelable:Boolean;
```

```
currentTarget:Boolean;
```

```
eventPhase:uint;
```

```
target:Object;
```

```
type:String;
```

# flash.events.Event

bubbles:Boolean;

cancelable:Boolean;

currentTarget:Boolean;

eventPhase:uint;

target:Object;

type:String;

**Индикира дали това събитие ще  
изплува нагоре**

# flash.events.Event

bubbles:Boolean;

cancelable:Boolean;

currentTarget:Boolean;

eventPhase:uint;

target:Object;

type:String;

**Индикира дали поведението асоциирано с това събитие може да бъде предотвратено**

# flash.events.Event

bubbles:Boolean;

cancelable:Boolean;

currentTarget:Boolean;

eventPhase:uint;

target:Object;

type:String;

**Обектът, чийто слушател в момента активно обработва събитието**

# flash.events.Event

```
bubbles:Boolean;
```

```
cancelable:Boolean;
```

```
currentTarget:Boolean;
```

```
eventPhase:uint;
```

```
target:Object;
```

```
type:String;
```

Текущата фаза в потока на  
обработката на събитието  
(*event flow*):

- **EventPhase.CAPTURING\_PHASE**
- **EventPhase.AT\_TARGET**
- **EventPhase.BUBBLING\_PHASE**



# flash.events.Event

bubbles:Boolean;

cancelable:Boolean;

currentTarget:Boolean;

eventPhase:uint;

target:Object;

type:String;

**Обектът излъчил събитието**

# flash.events.Event

bubbles:Boolean;

cancelable:Boolean;

currentTarget:Boolean;

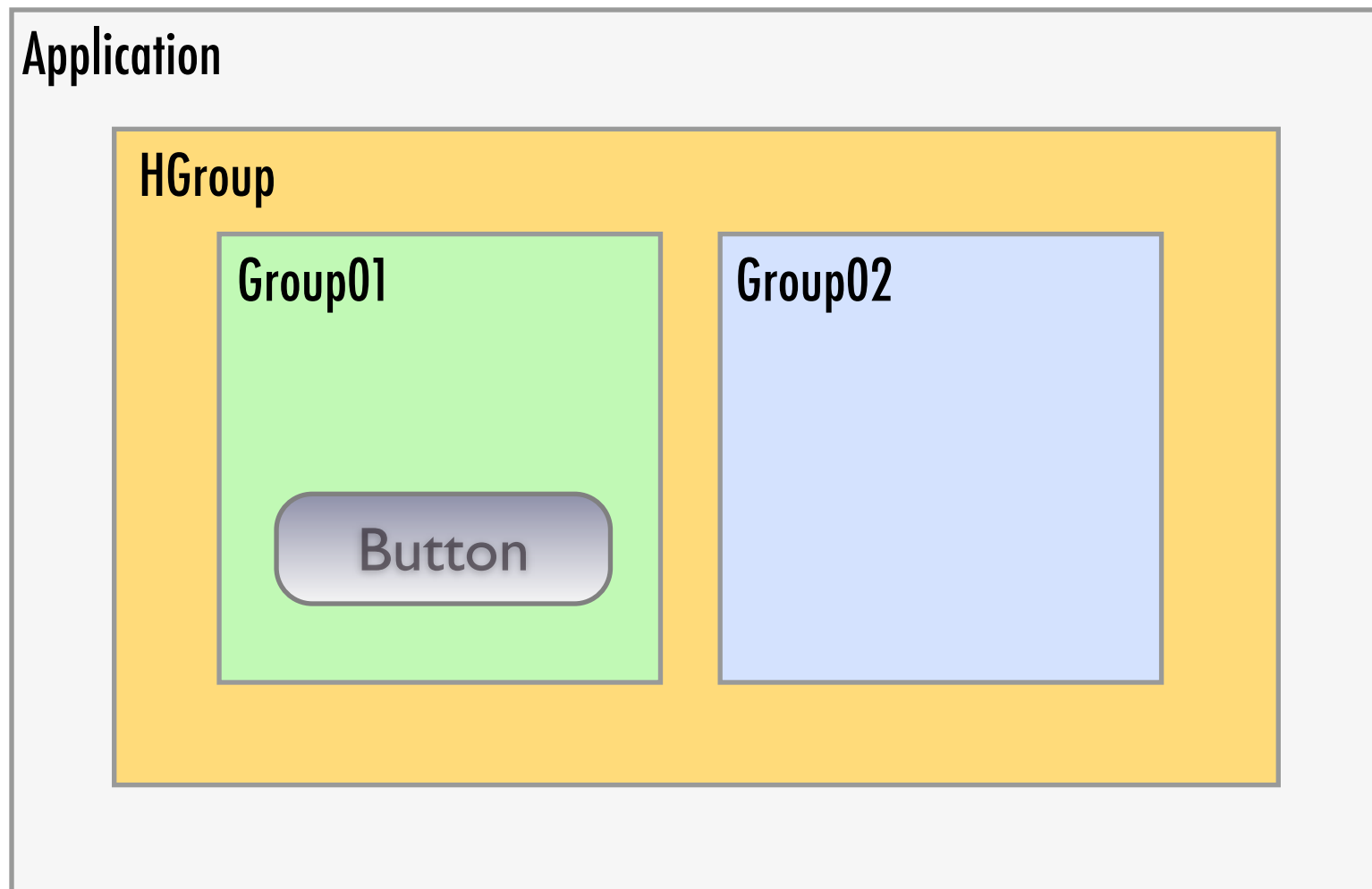
eventPhase:uint;

target:Object;

type:String;

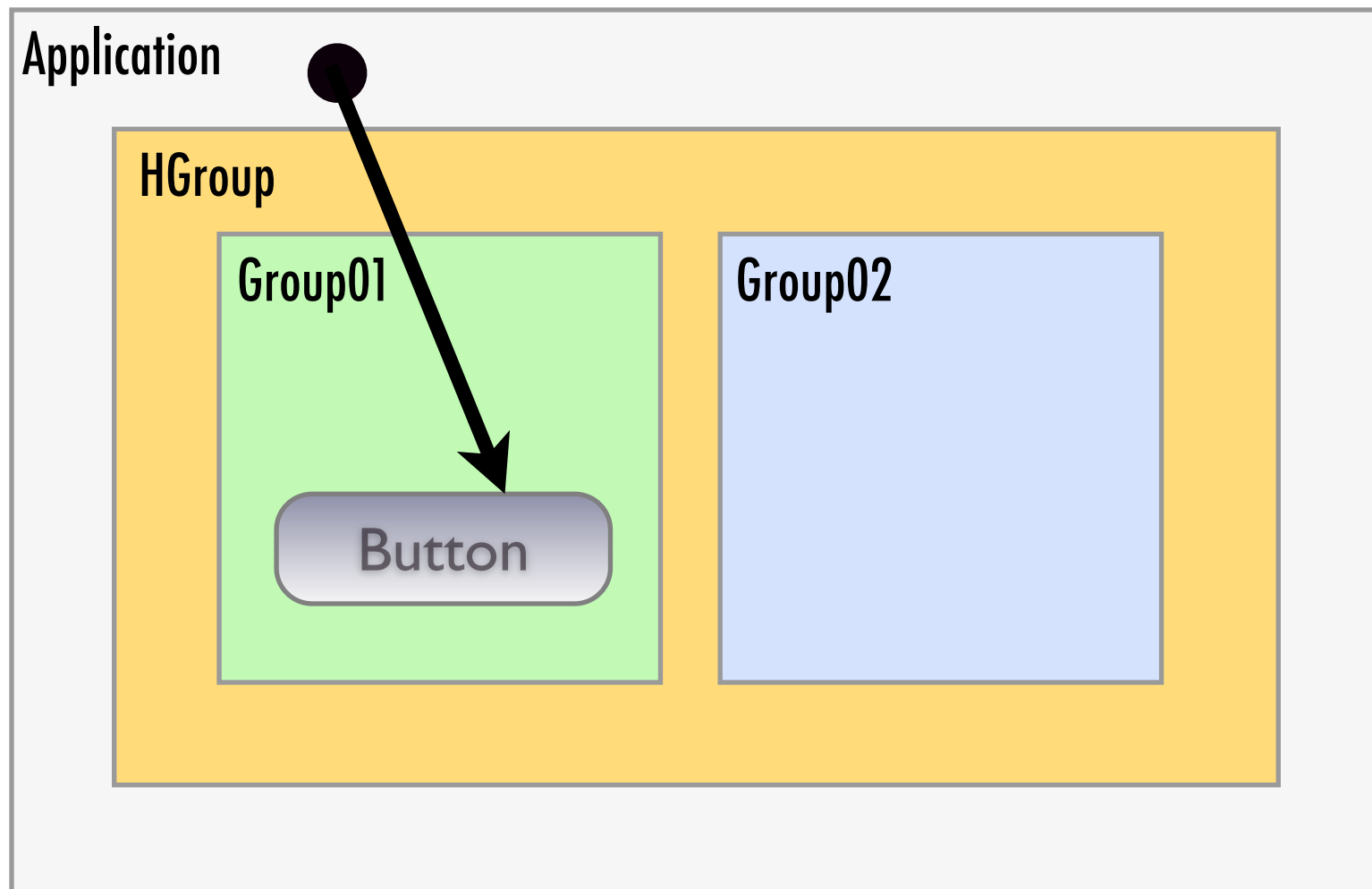
**Името на събитието е неговият тип**  
**- главни и малки буква са от**  
**значение**

# Event Propagation



- *display list* - йерархичен контейнер на всички визуални обекти
- когато събитието настъпва, то сякаш настъпва за цялата *display list*-а, а не само за клипа го излъчва

# Event Propagation

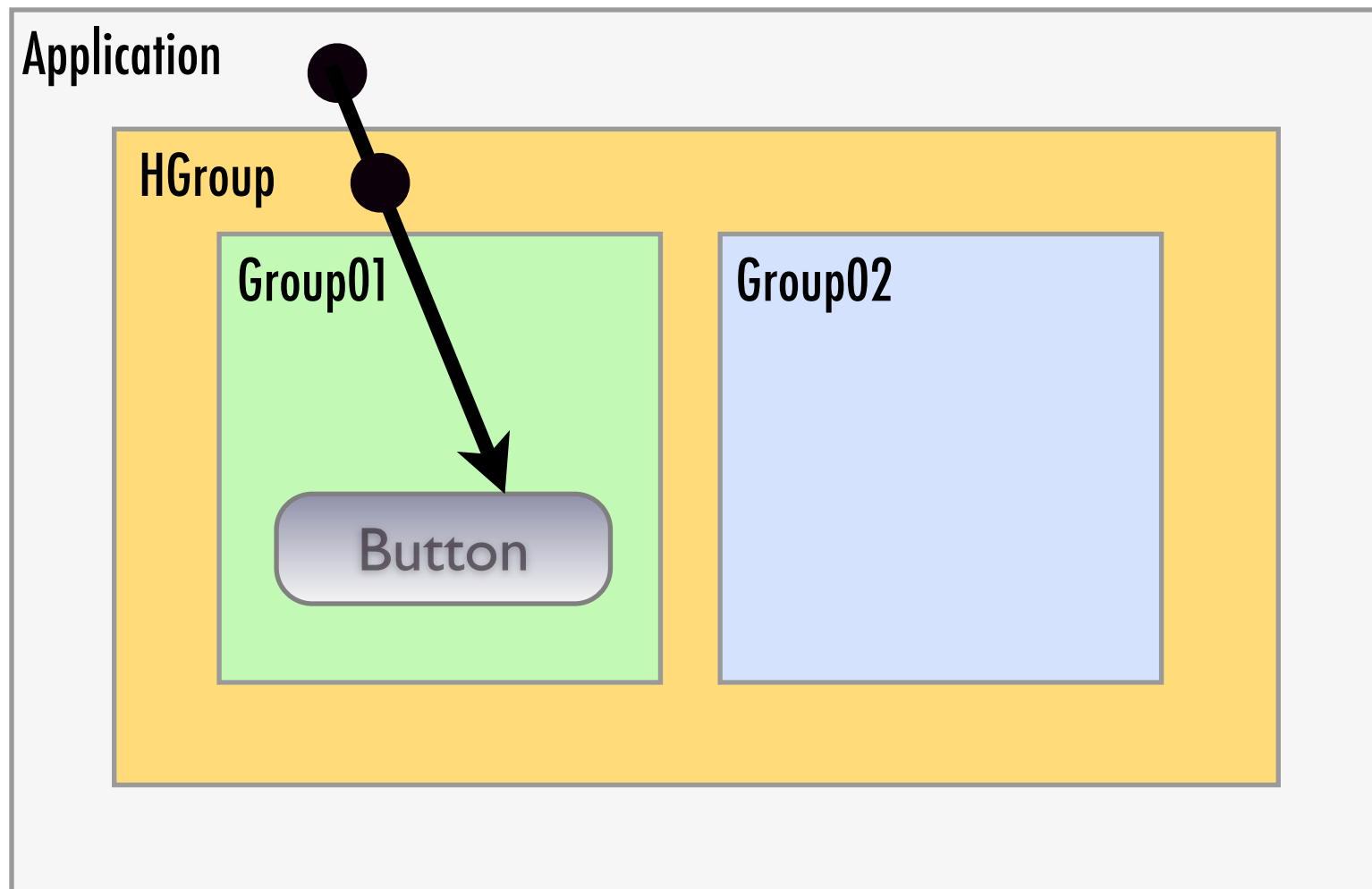


```
eventPhase = EventPhase.CAPTURING_PHASE;
```

```
target = button;
```

```
currentTarget = application;
```

# Event Propagation

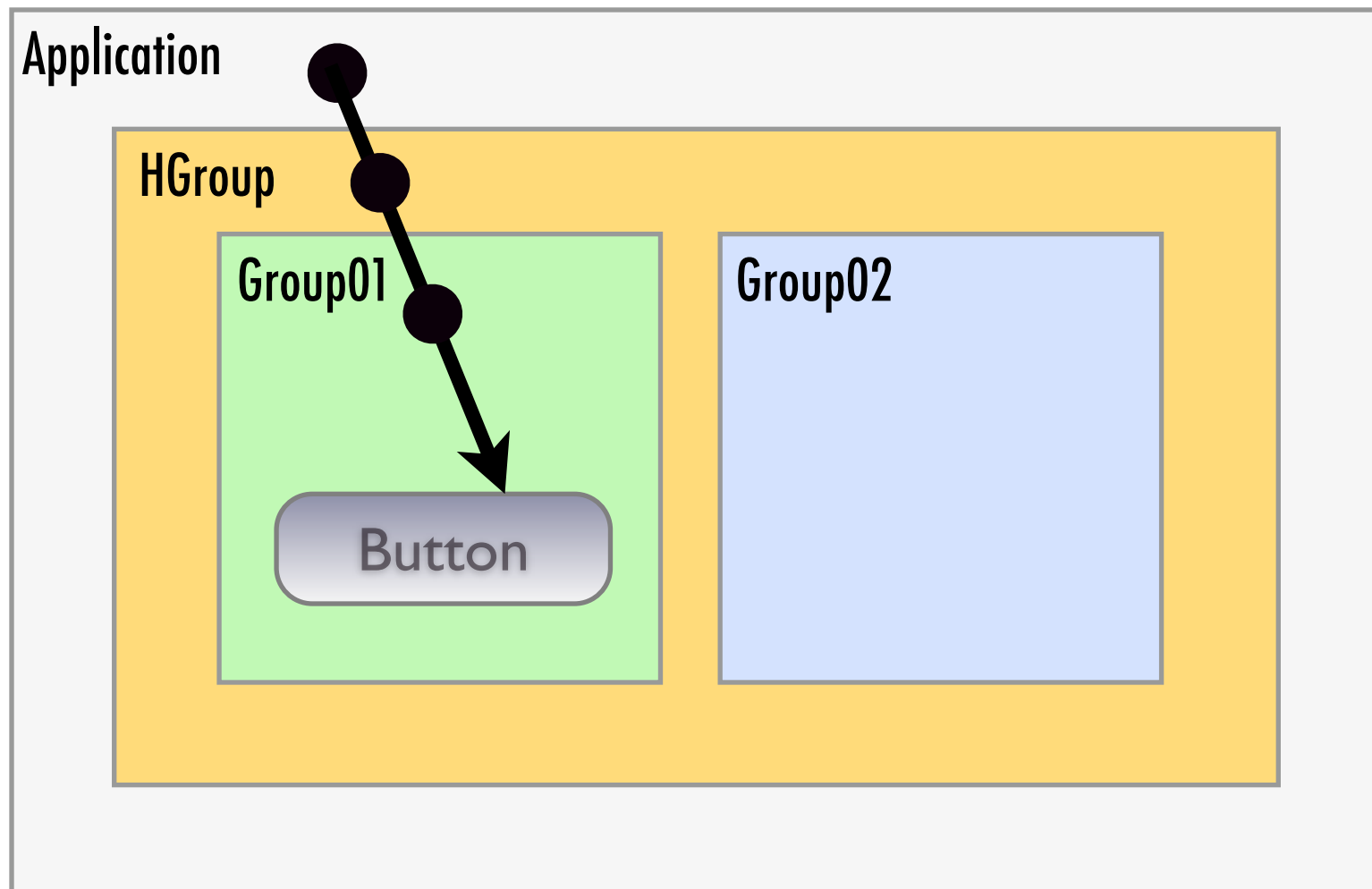


```
eventPhase = EventPhase.CAPTURING_PHASE;
```

```
target = button;
```

```
currentTarget = hgroup;
```

# Event Propagation

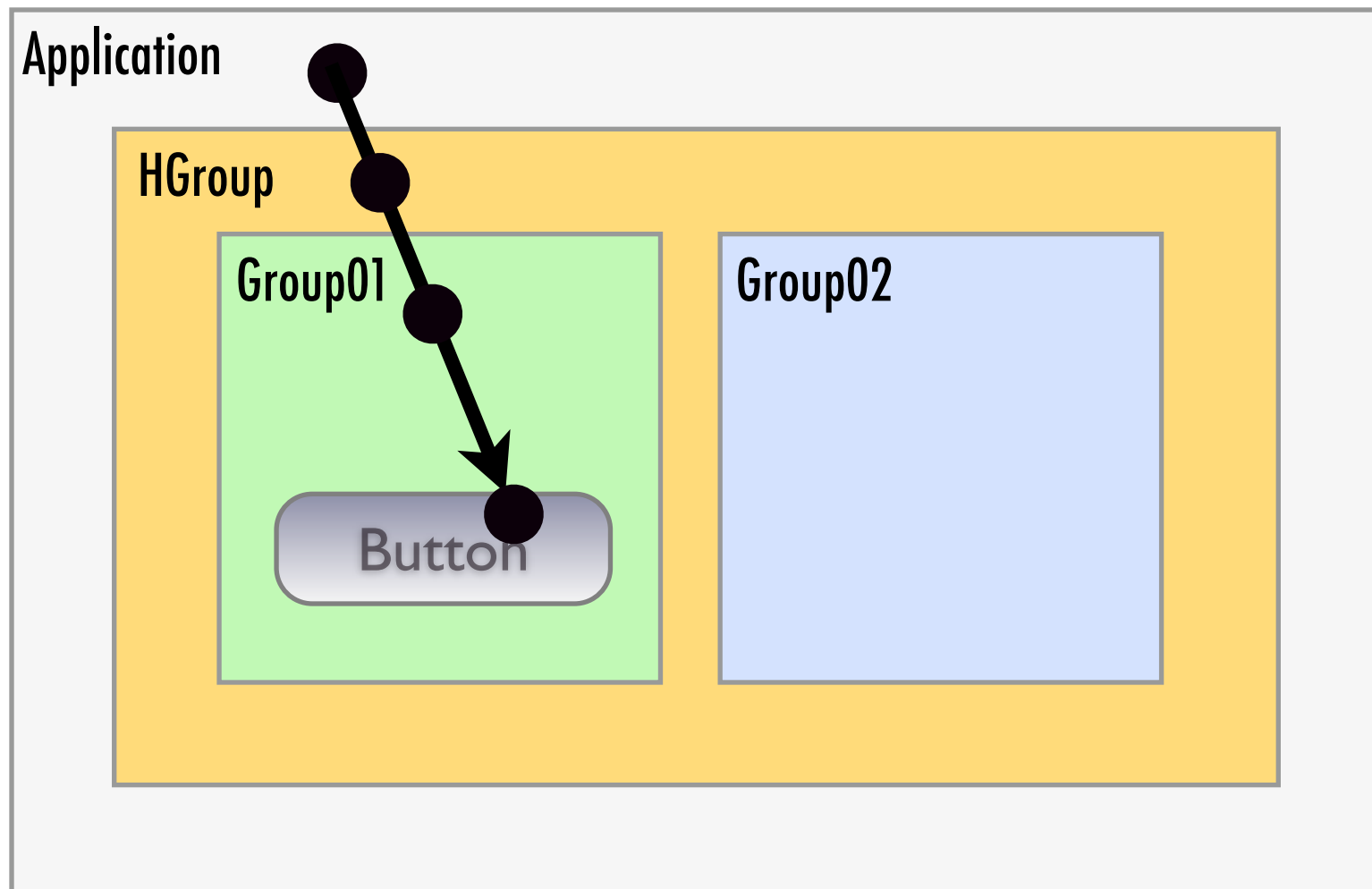


```
eventPhase = EventPhase.CAPTURING_PHASE;
```

```
target = button;
```

```
currentTarget = group01;
```

# Event Propagation

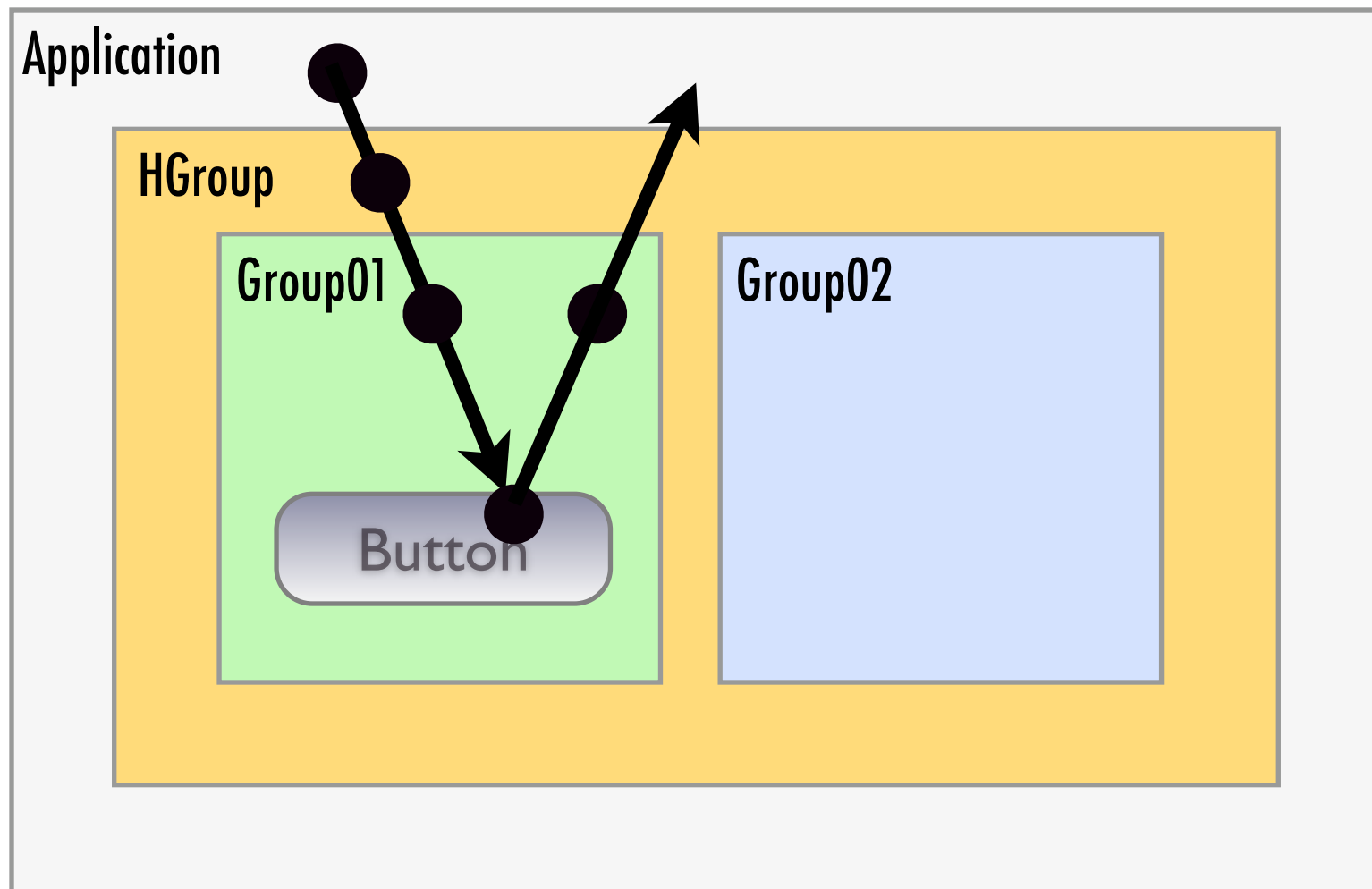


```
eventPhase = EventPhase.AT_TARGET;
```

```
target = button;
```

```
currentTarget = button;
```

# Event Propagation



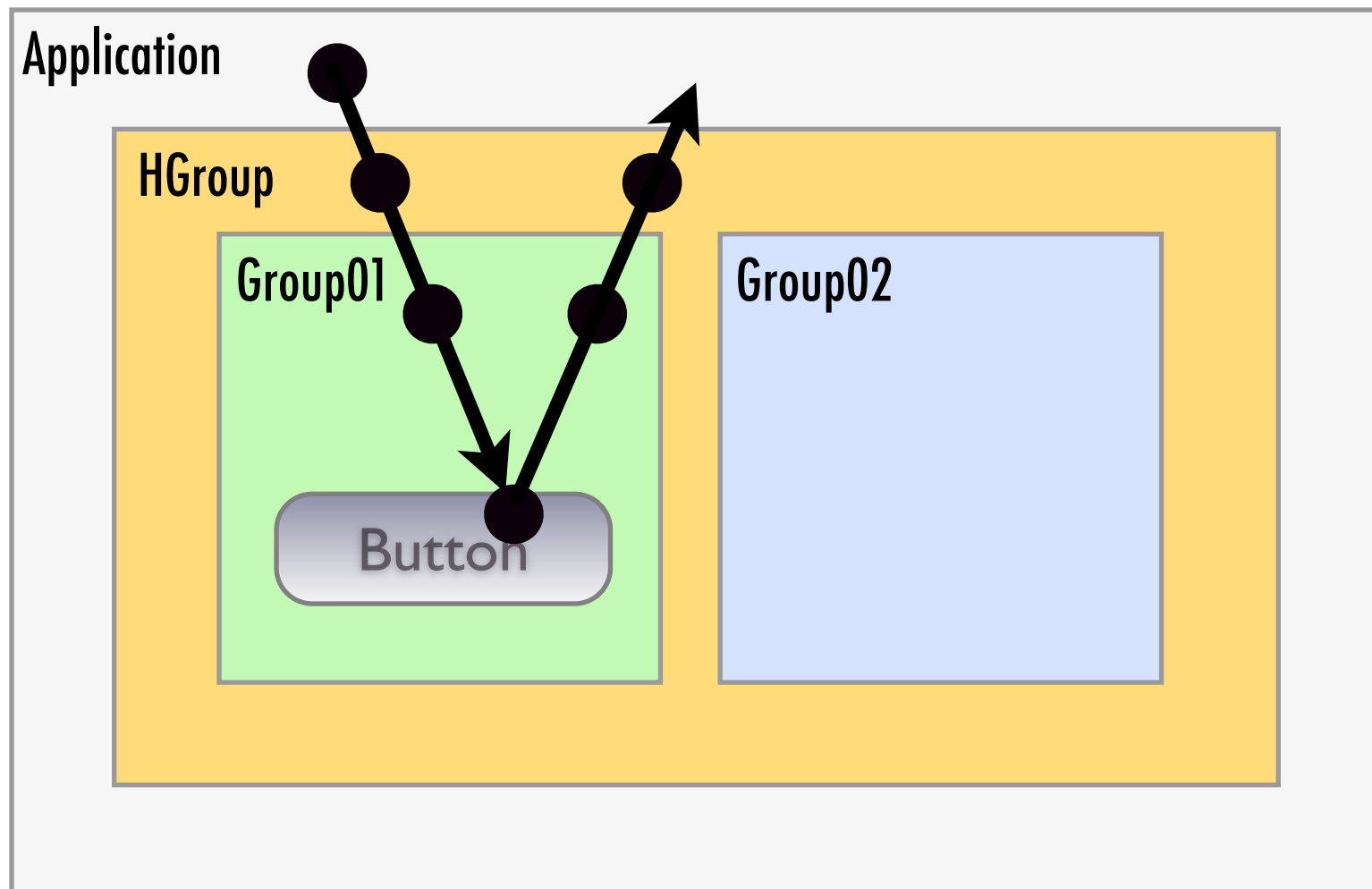
```
eventPhase = EventPhase.BUBBLING_PHASE;
```

```
target = button;
```

```
currentTarget = group01;
```



# Event Propagation

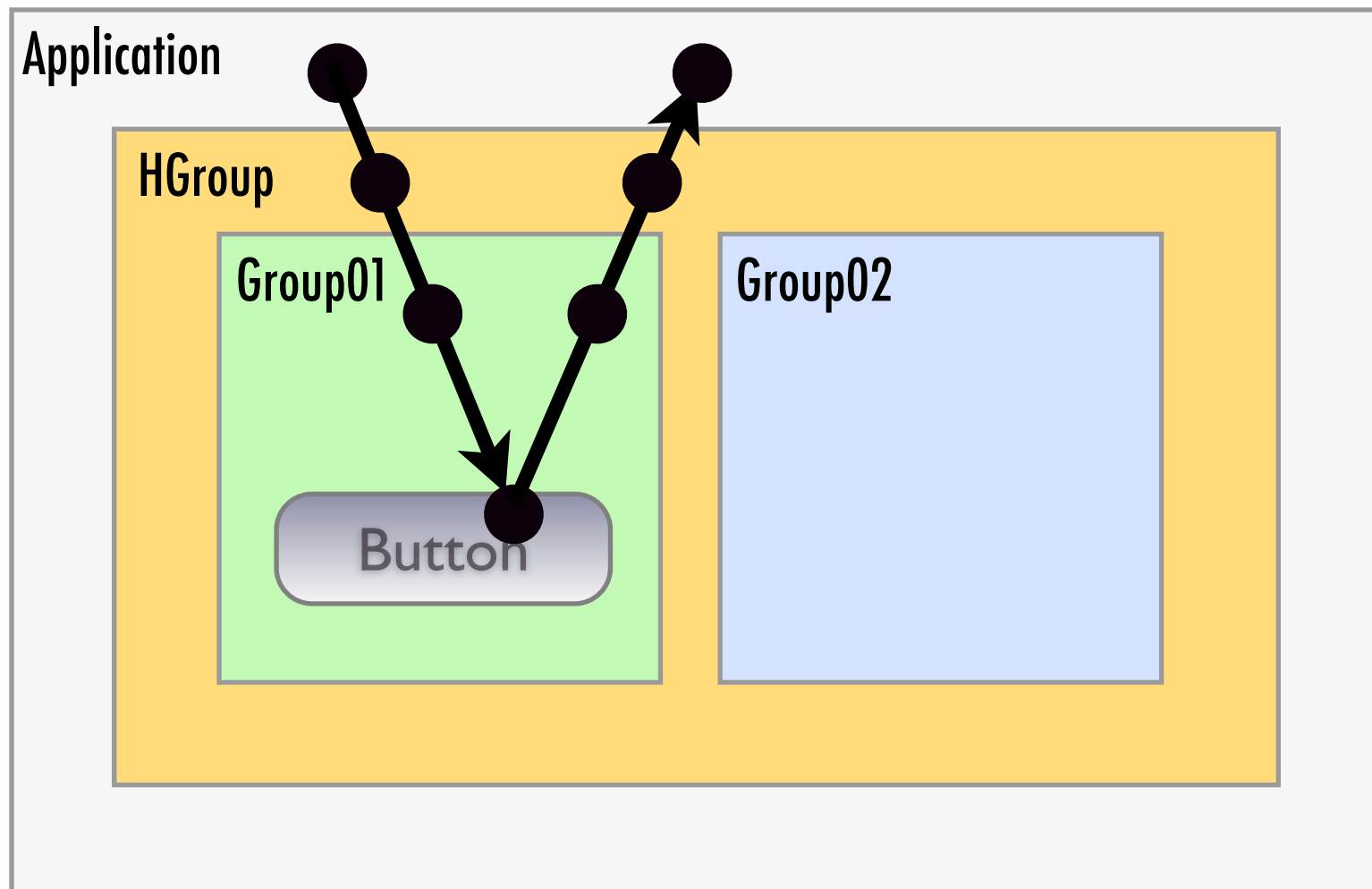


```
eventPhase = EventPhase.BUBBLING_PHASE;
```

```
target = button;
```

```
currentTarget = hgroup;
```

# Event Propagation



```
eventPhase = EventPhase.BUBBLING_PHASE;
```

```
target = button;
```

```
currentTarget = application;
```

```
event.stopPropagation();
```

- **във всяка фаза от потока на събитието ние можем да спрем неговото пропагандиране и така да предотвратим изпълнението на слушателите нагоре по веригата**
- **позволява да се изпълнят слушателите на текущото ниво**

```
event.stopImmediatePropagation();
```

- **предотвратява изпълнението на всички слушатели, включително и тези на текущото ниво**

# Обобщение

- **ВЪВЕДЕНИЕ В event-ите**
- **flash.events.IEventDispatcher**
- **flash.events.Event**
- **event propagation**
- **преустановяване на събитие**