

Long Distance Coach  
長距離バス 解説

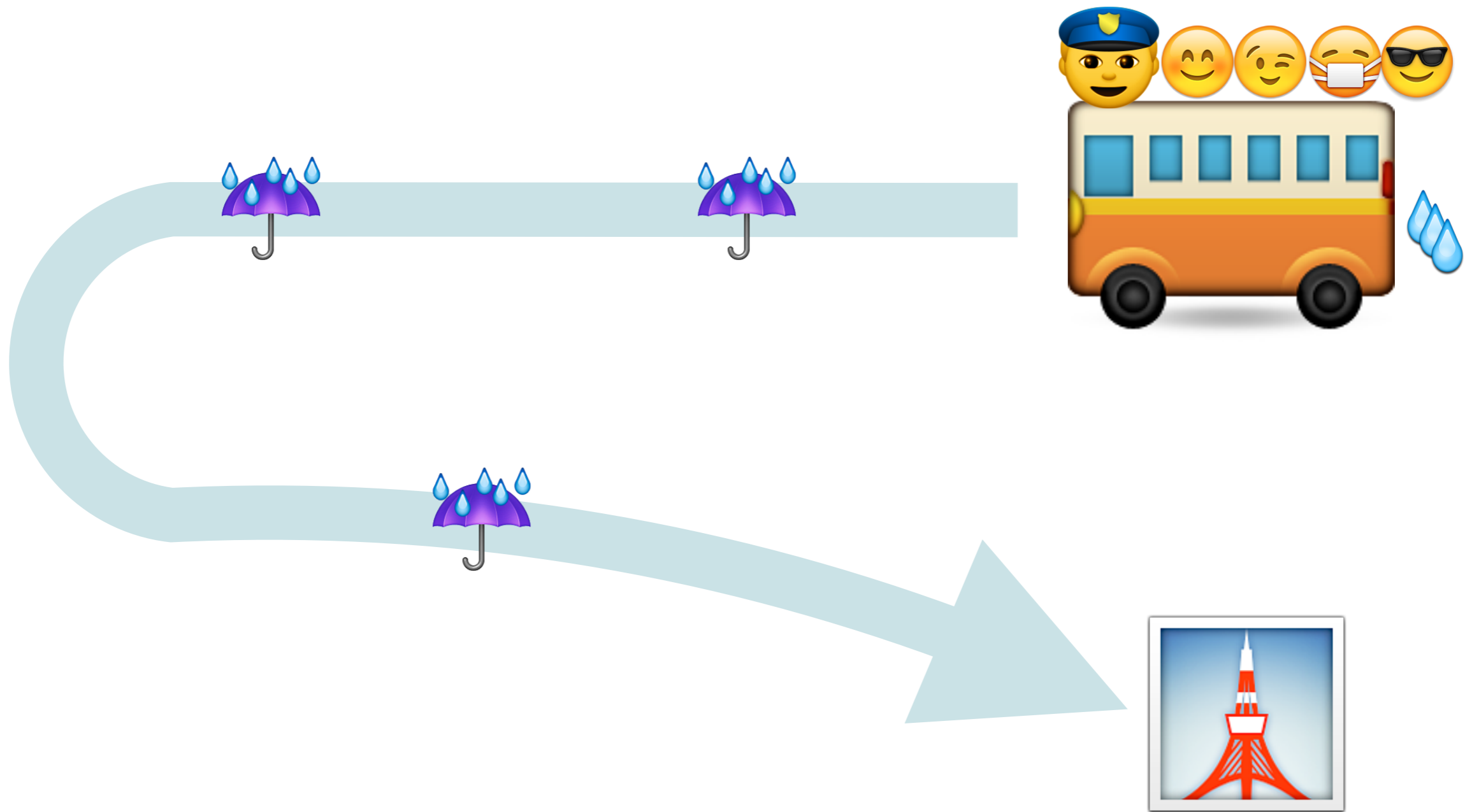
---

今西 健介 (@japlj)

情報オリンピック 2016-2017 春季トレーニング合宿 競技日3

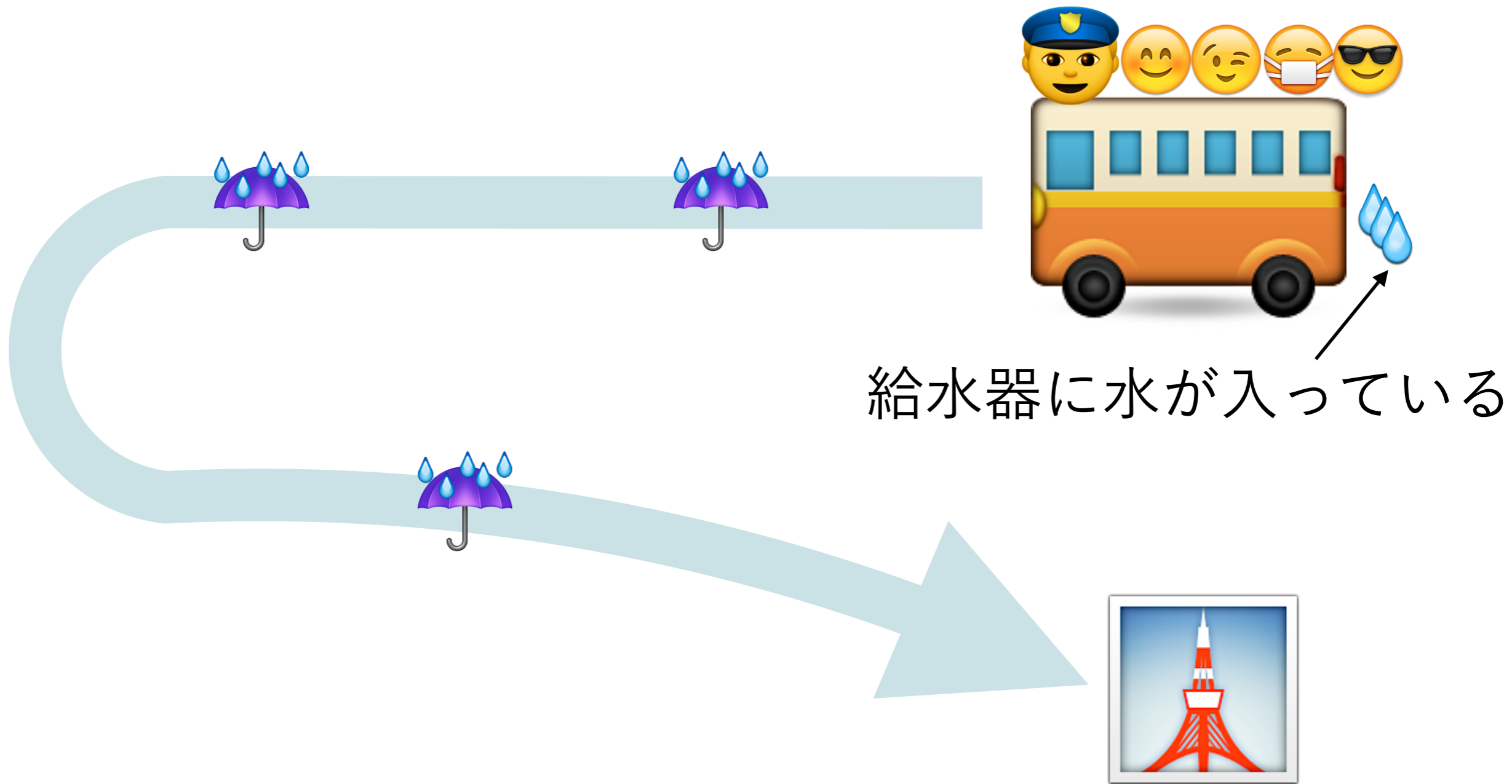
# 問題概要

長距離バスが目的地に行きたい



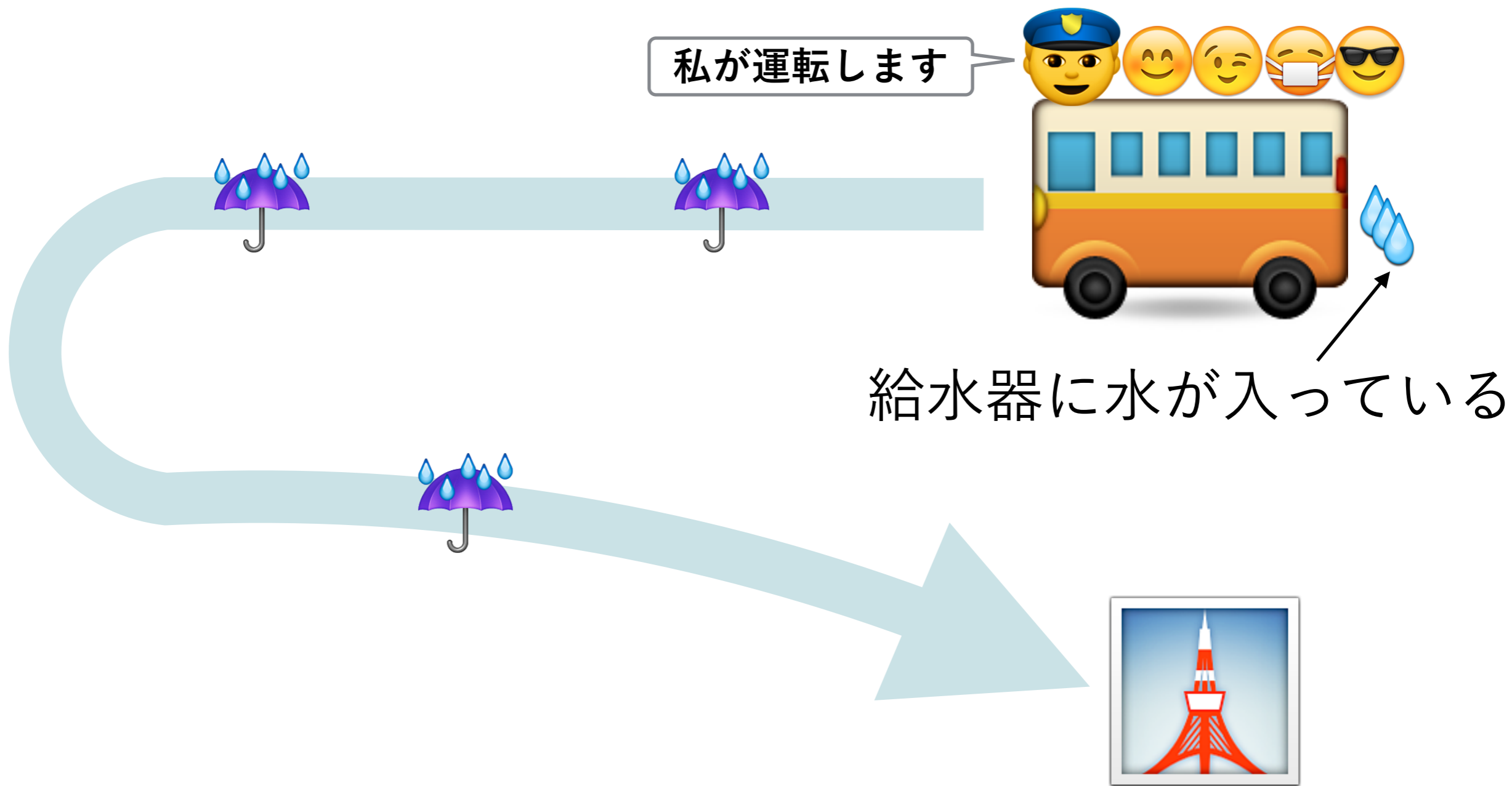
# 問題概要

長距離バスが目的地に行きたい



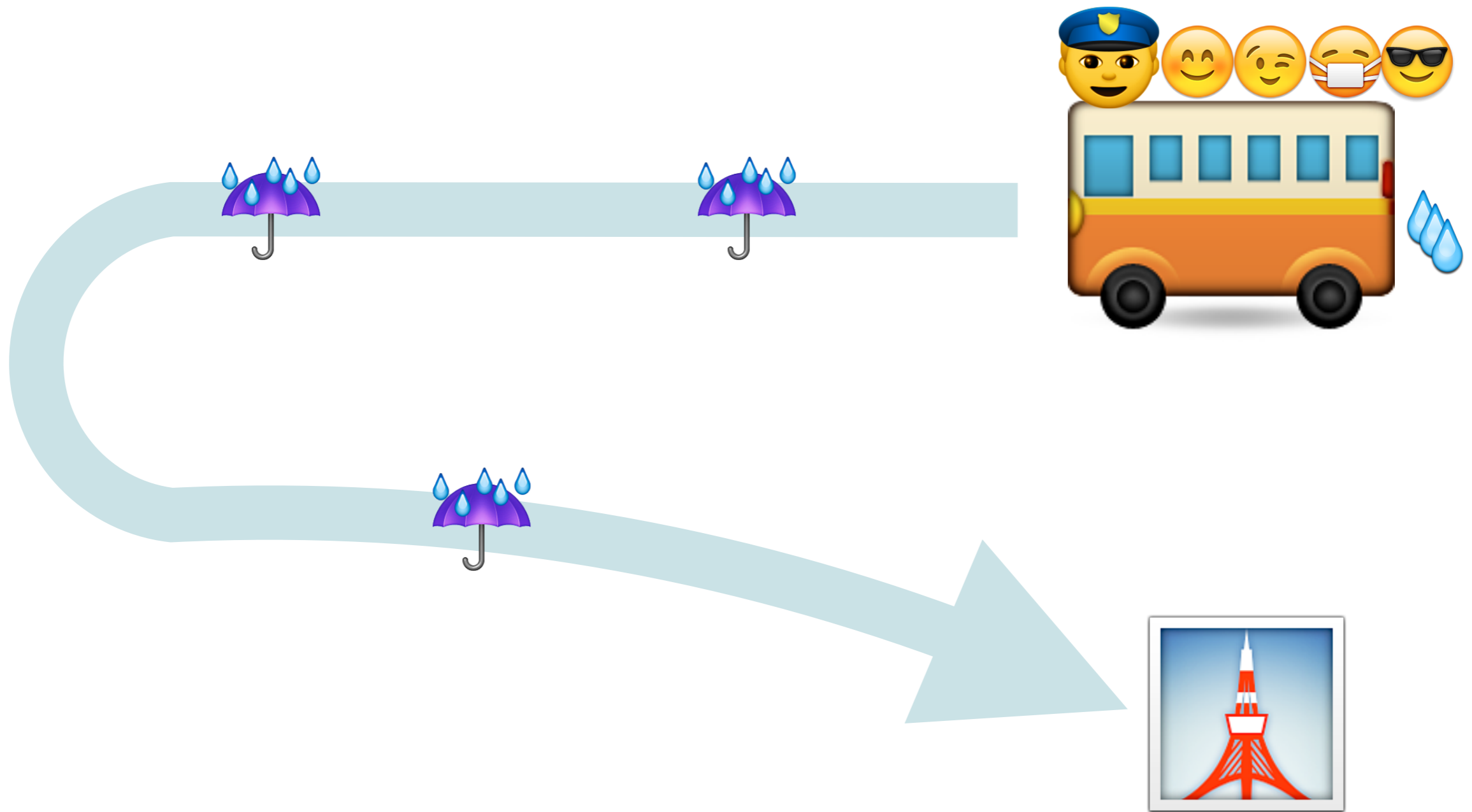
# 問題概要

長距離バスが目的地に行きたい



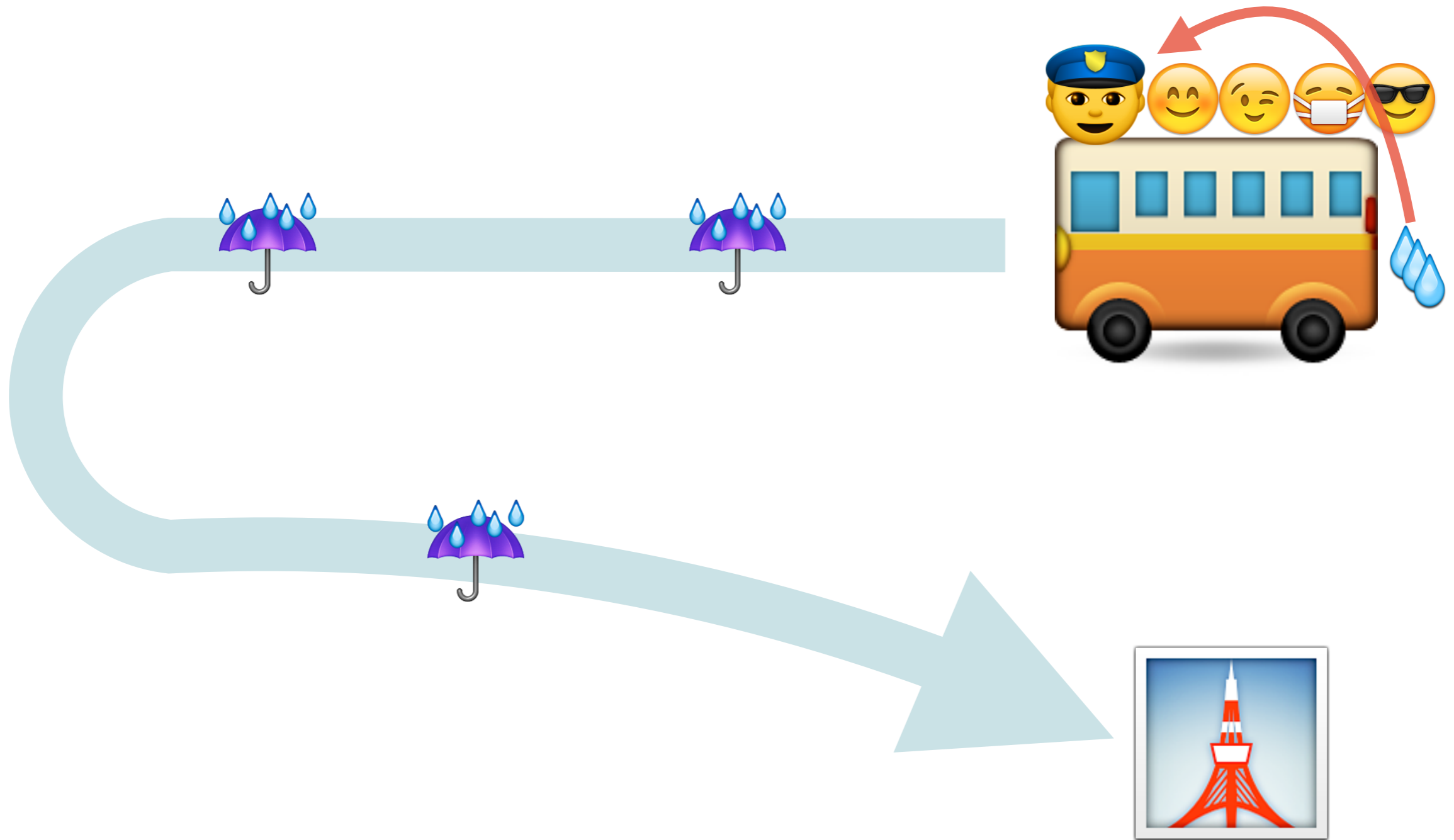
# 問題概要

長距離バスが目的地に行きたい



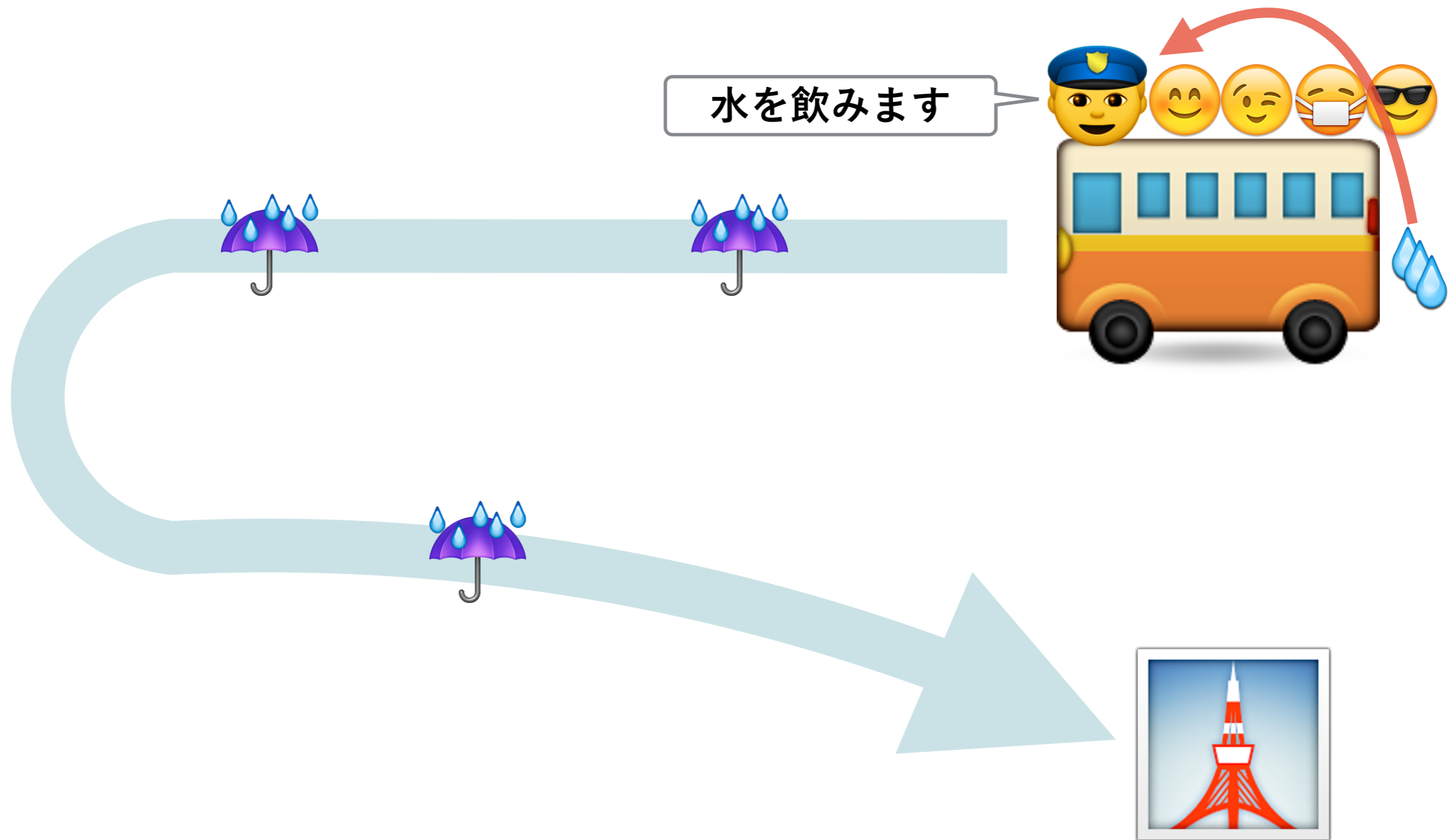
# 問題概要

長距離バスが目的地に行きたい



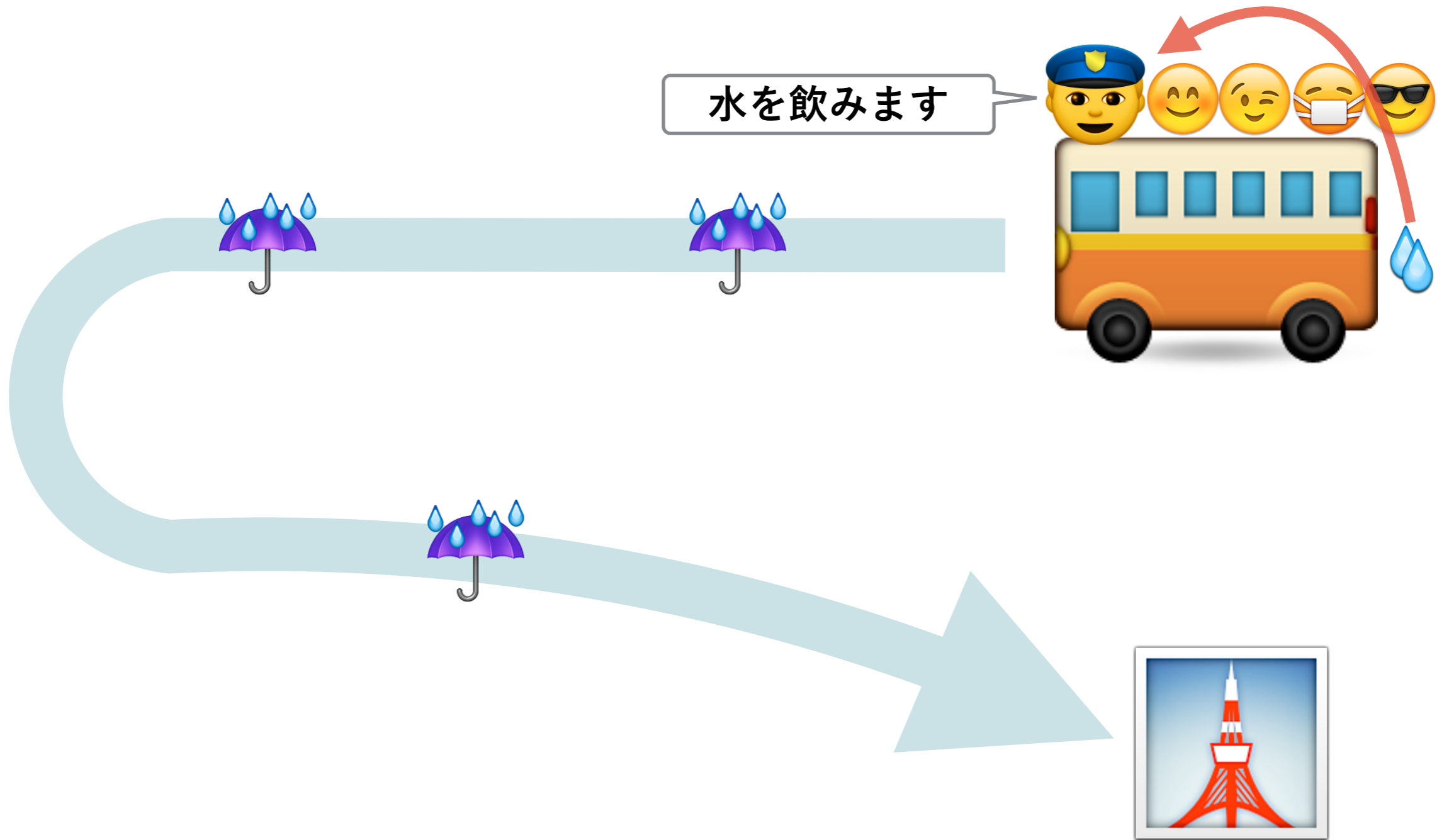
# 問題概要

長距離バスが目的地に行きたい



# 問題概要

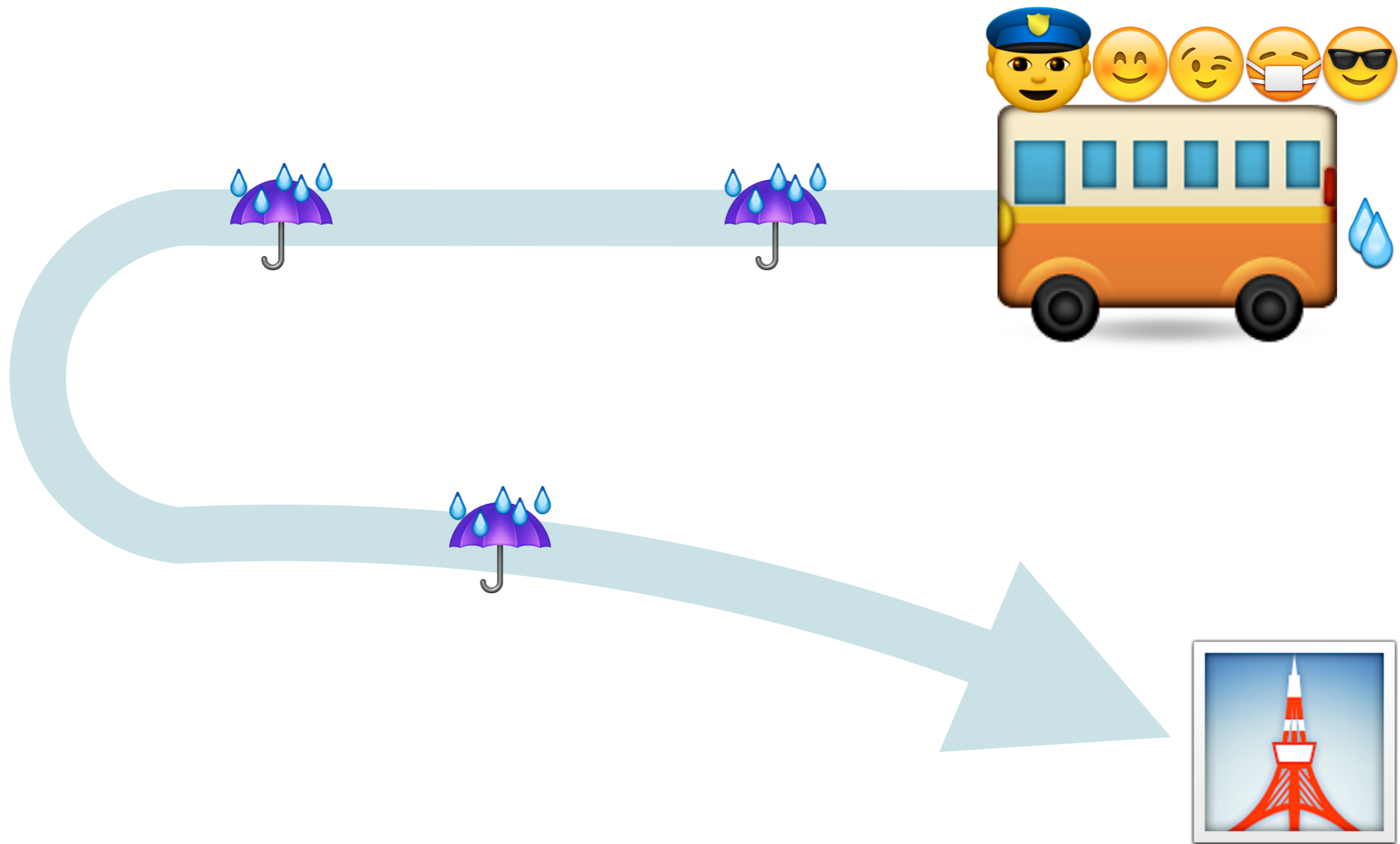
長距離バスが目的地に行きたい





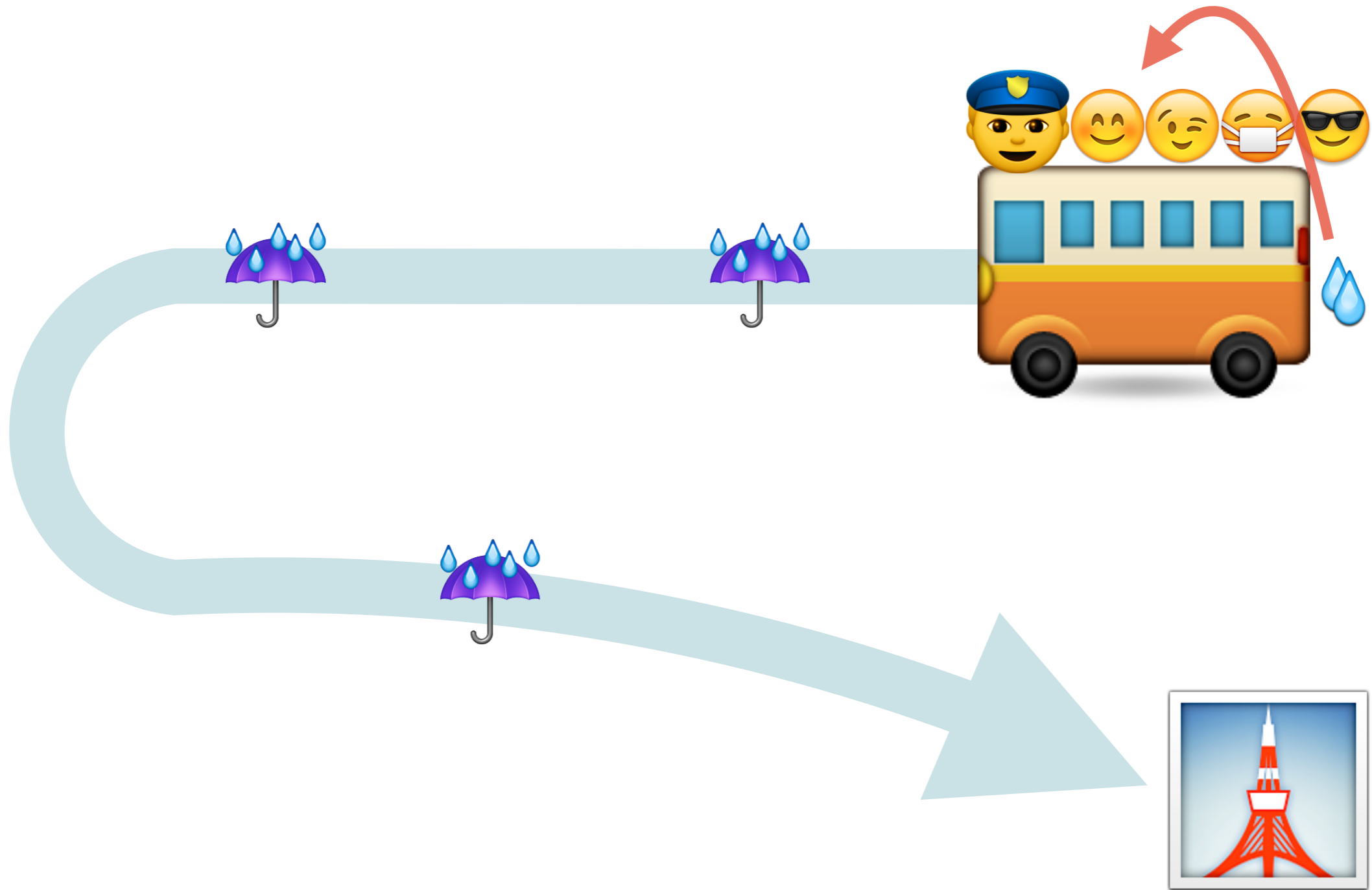
# 問題概要

長距離バスが目的地に行きたい



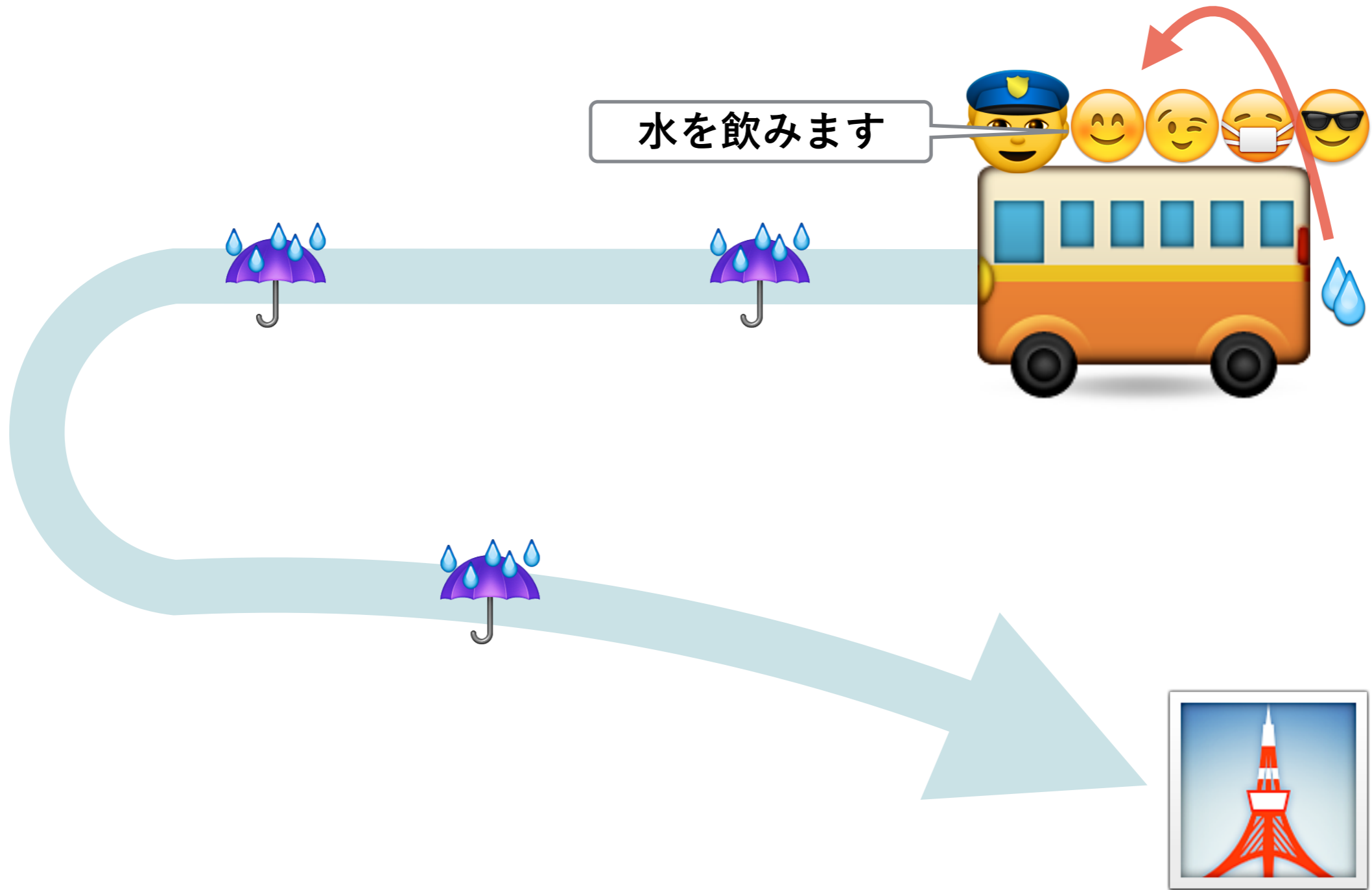
# 問題概要

長距離バスが目的地に行きたい



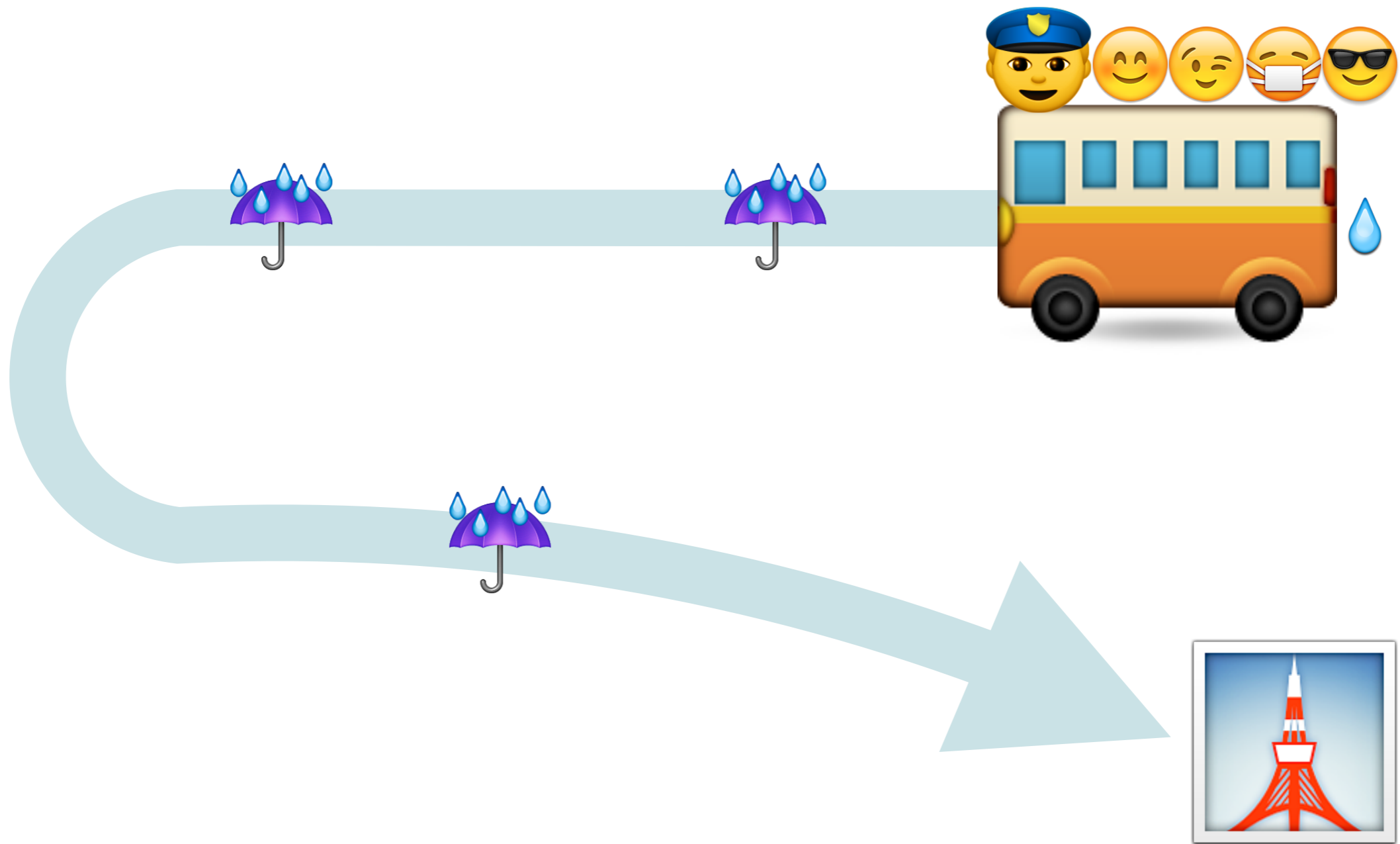
# 問題概要

長距離バスが目的地に行きたい



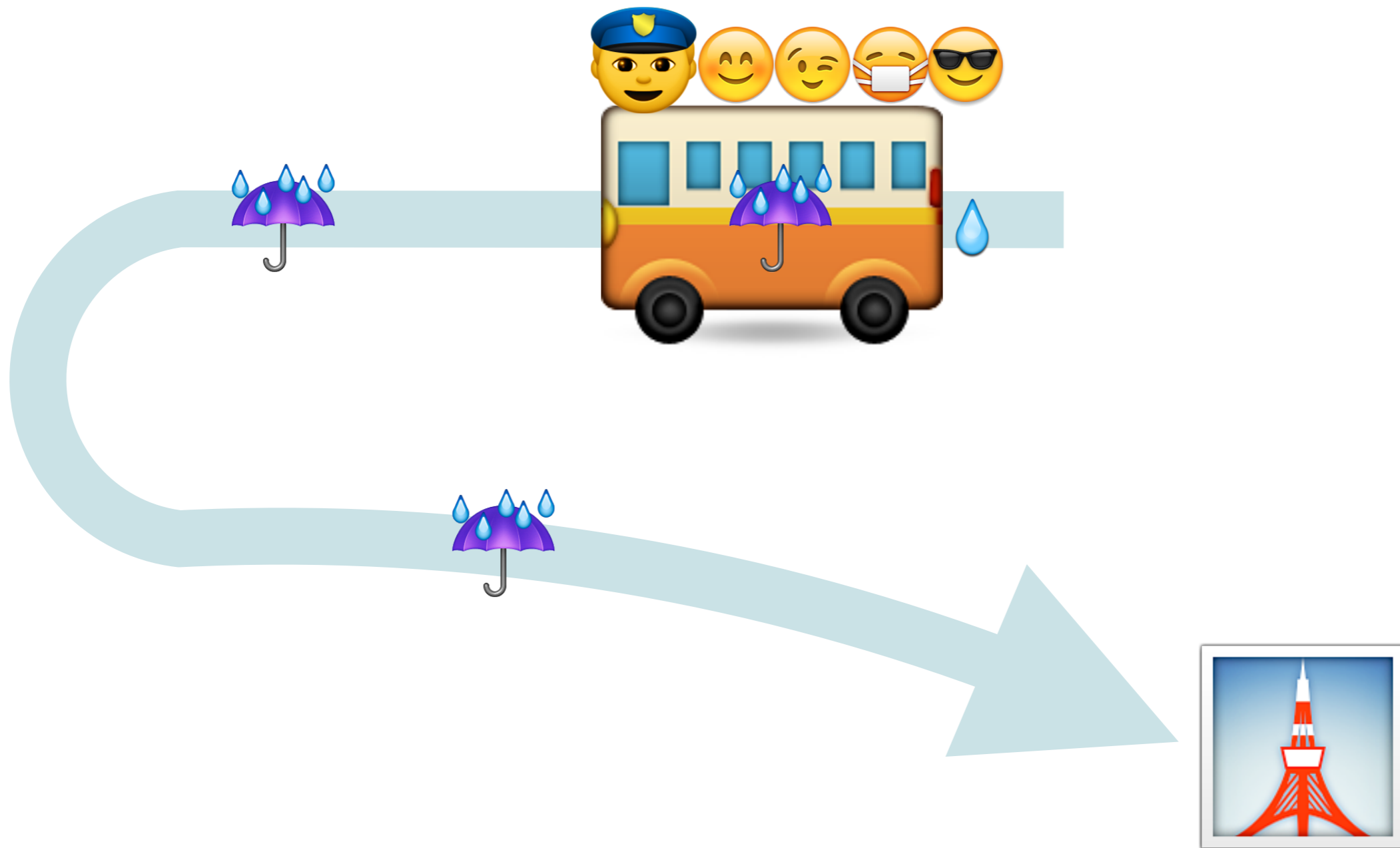
# 問題概要

長距離バスが目的地に行きたい



# 問題概要

長距離バスが目的地に行きたい



# 問題概要

長距離バスが目的地に行きたい



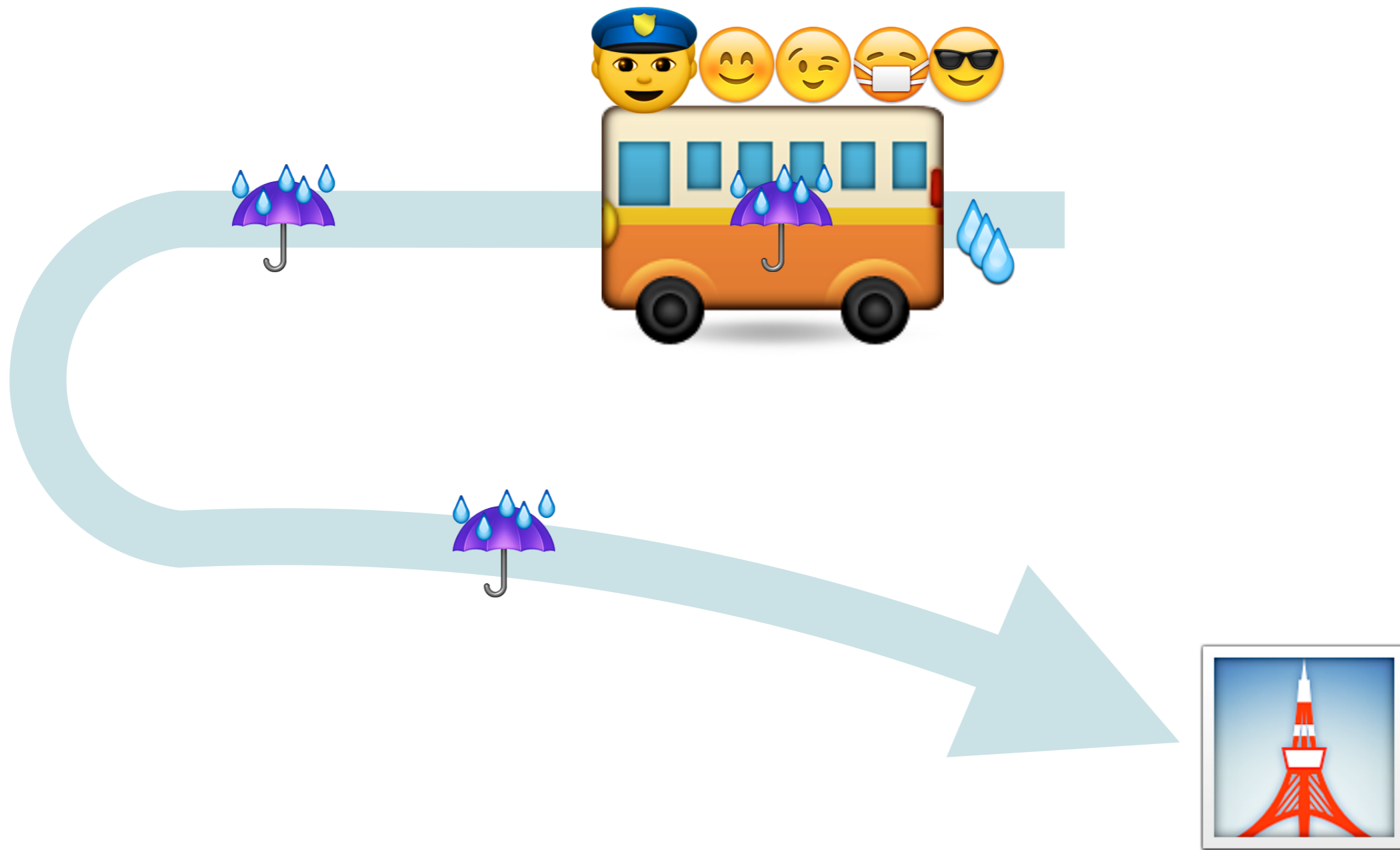
# 問題概要

長距離バスが目的地に行きたい



# 問題概要

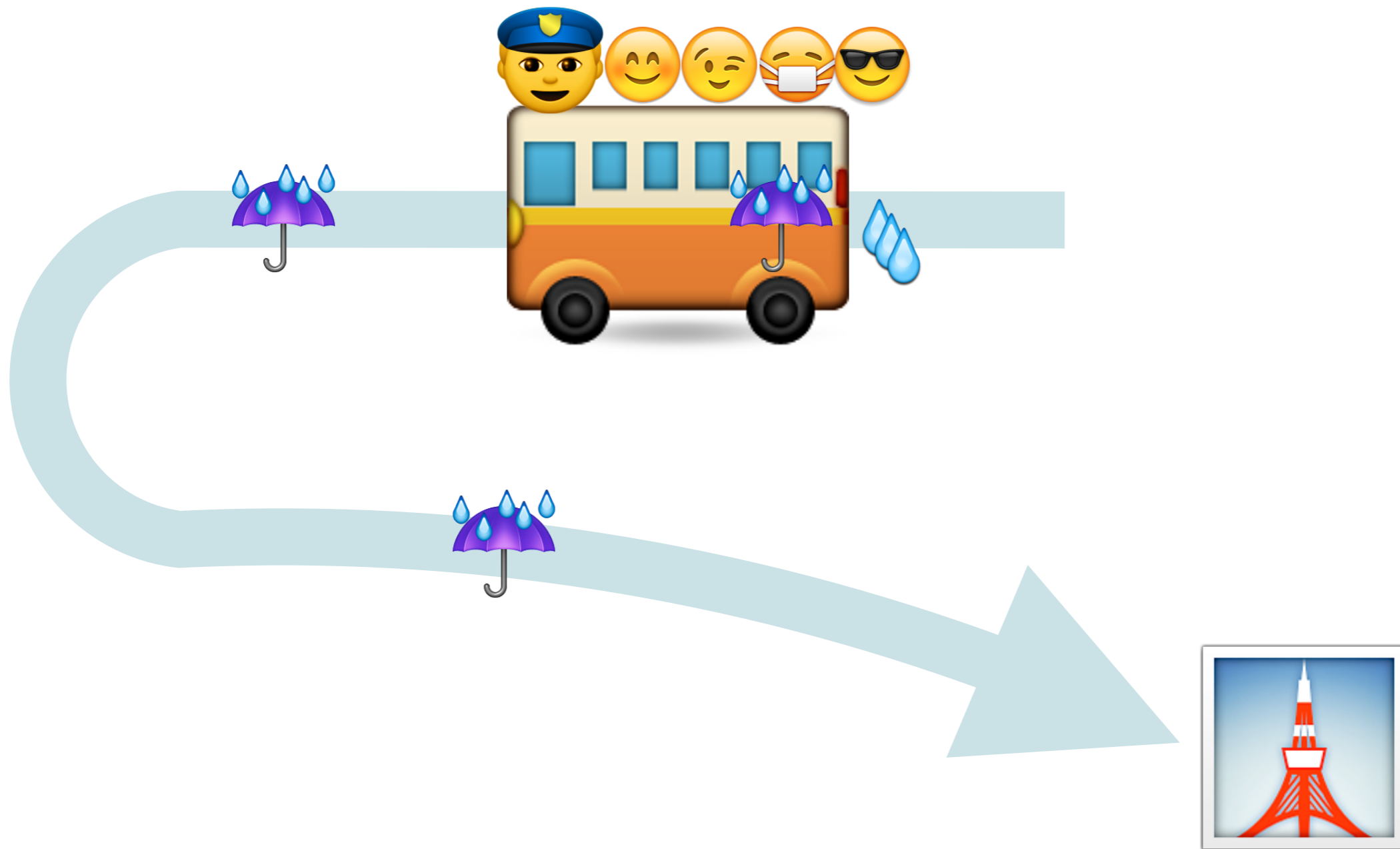
長距離バスが目的地に行きたい





# 問題概要

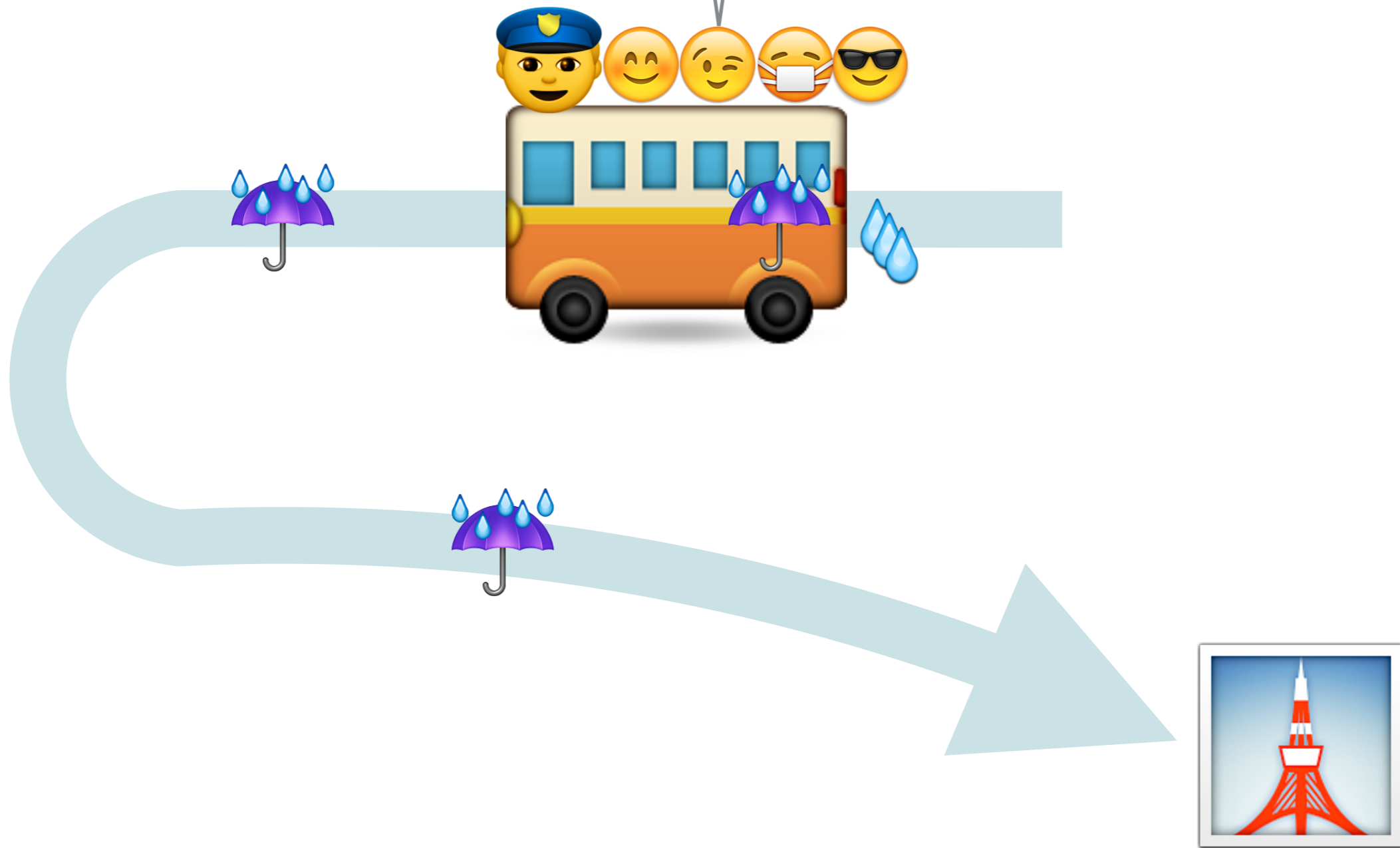
長距離バスが目的地に行きたい



# 問題概要

長距離バスが目的地に行きたい

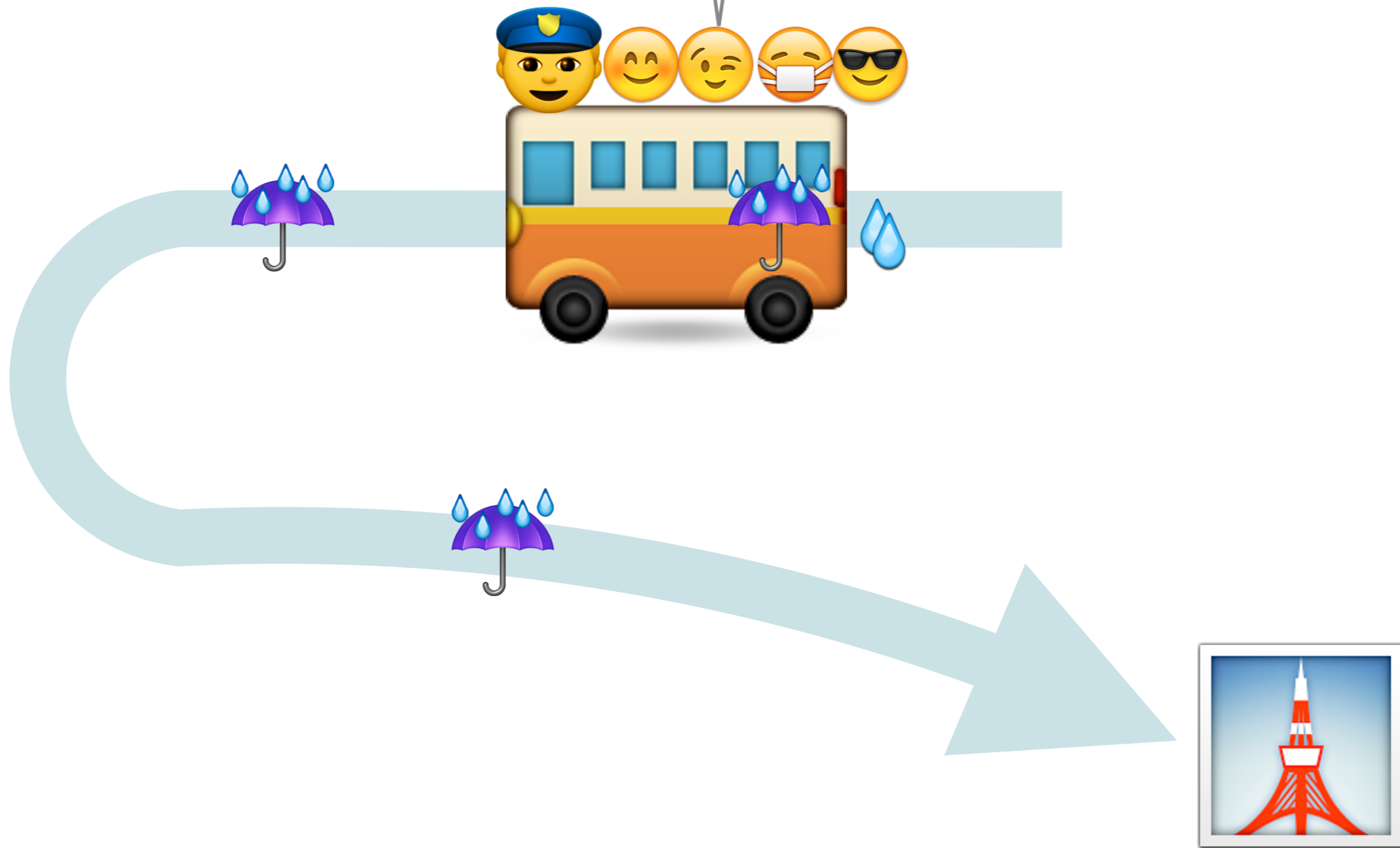
水を飲みます



# 問題概要

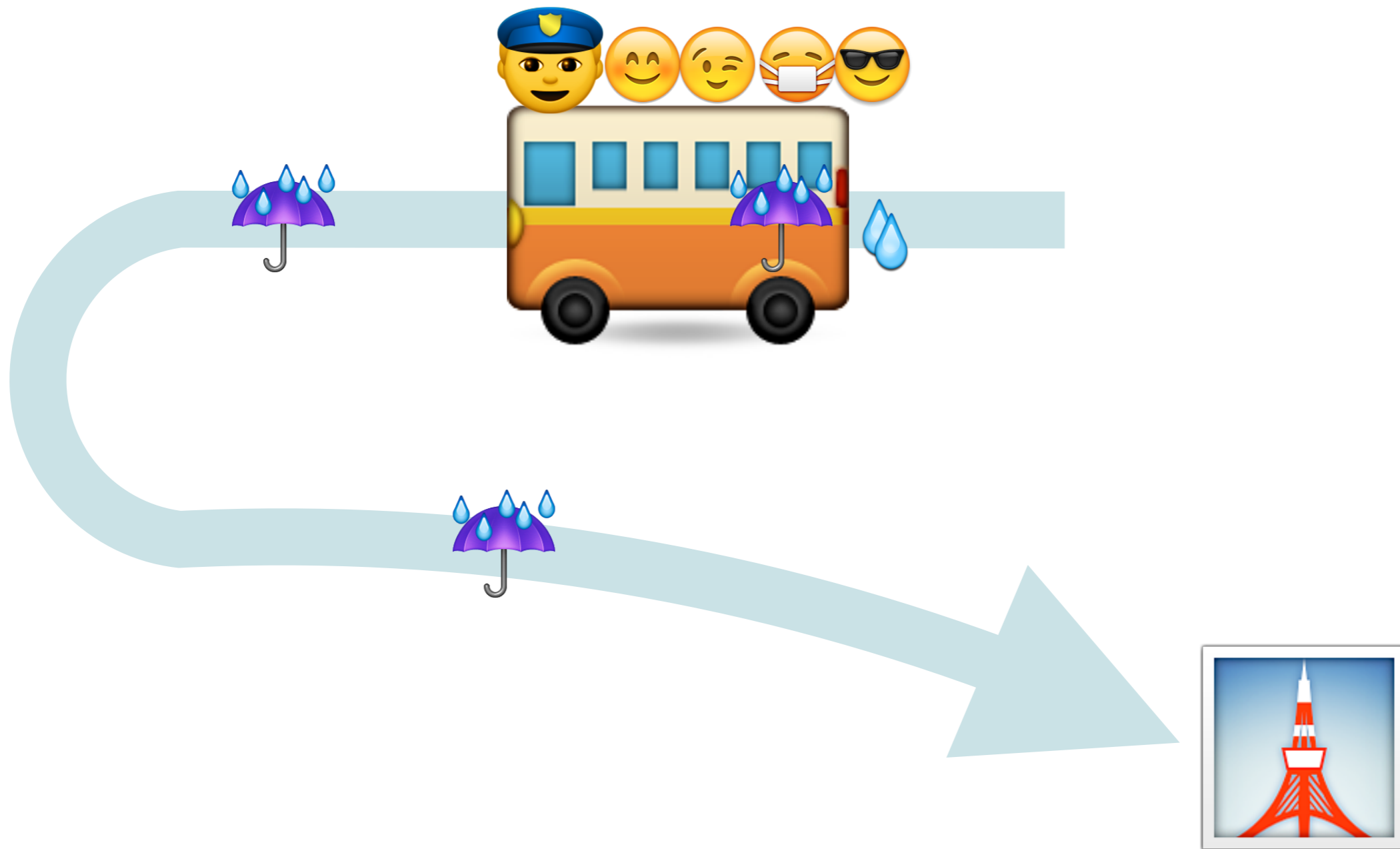
長距離バスが目的地に行きたい

水を飲みます



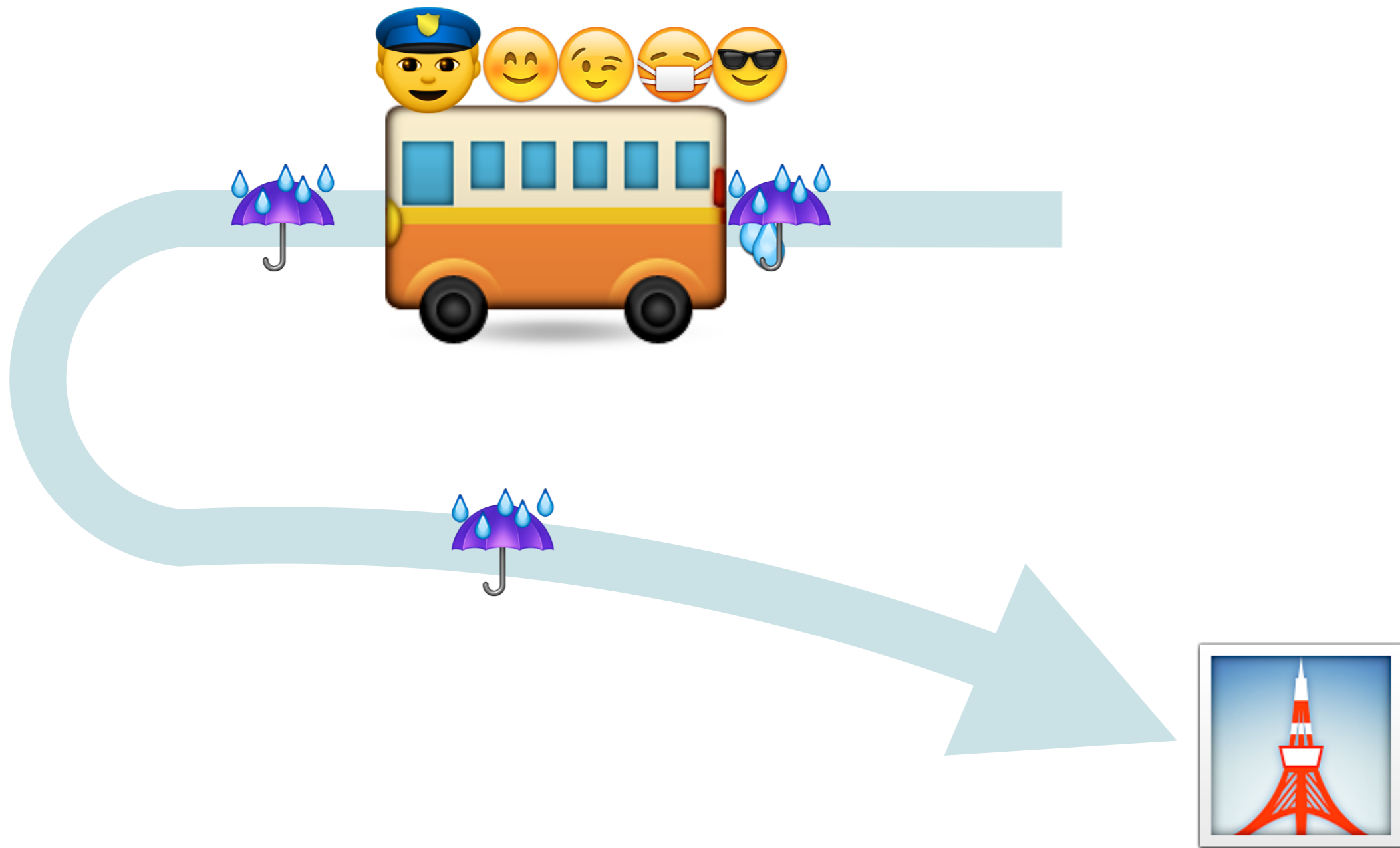
# 問題概要

長距離バスが目的地に行きたい



# 問題概要

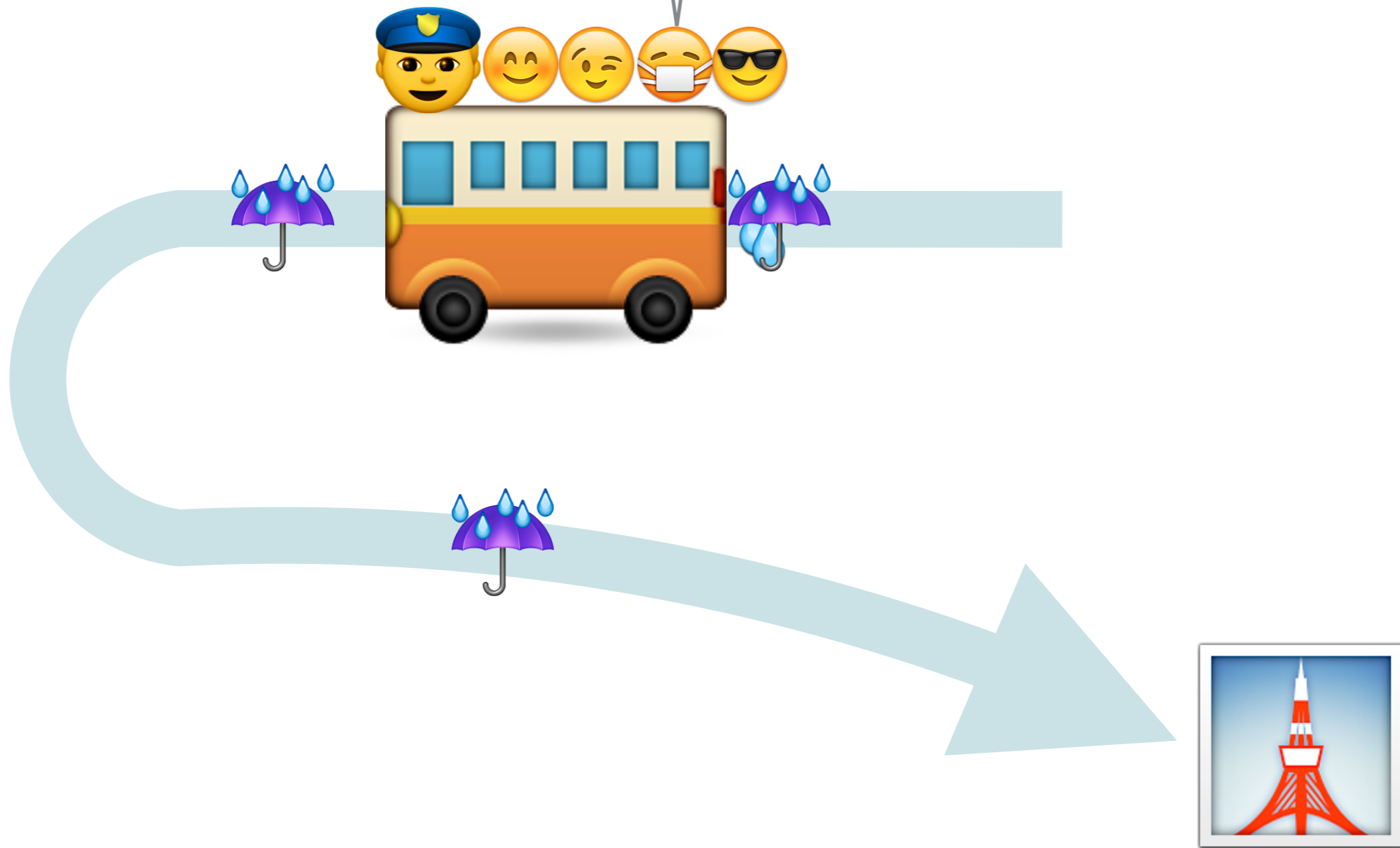
長距離バスが目的地に行きたい



# 問題概要

長距離バスが目的地に行きたい

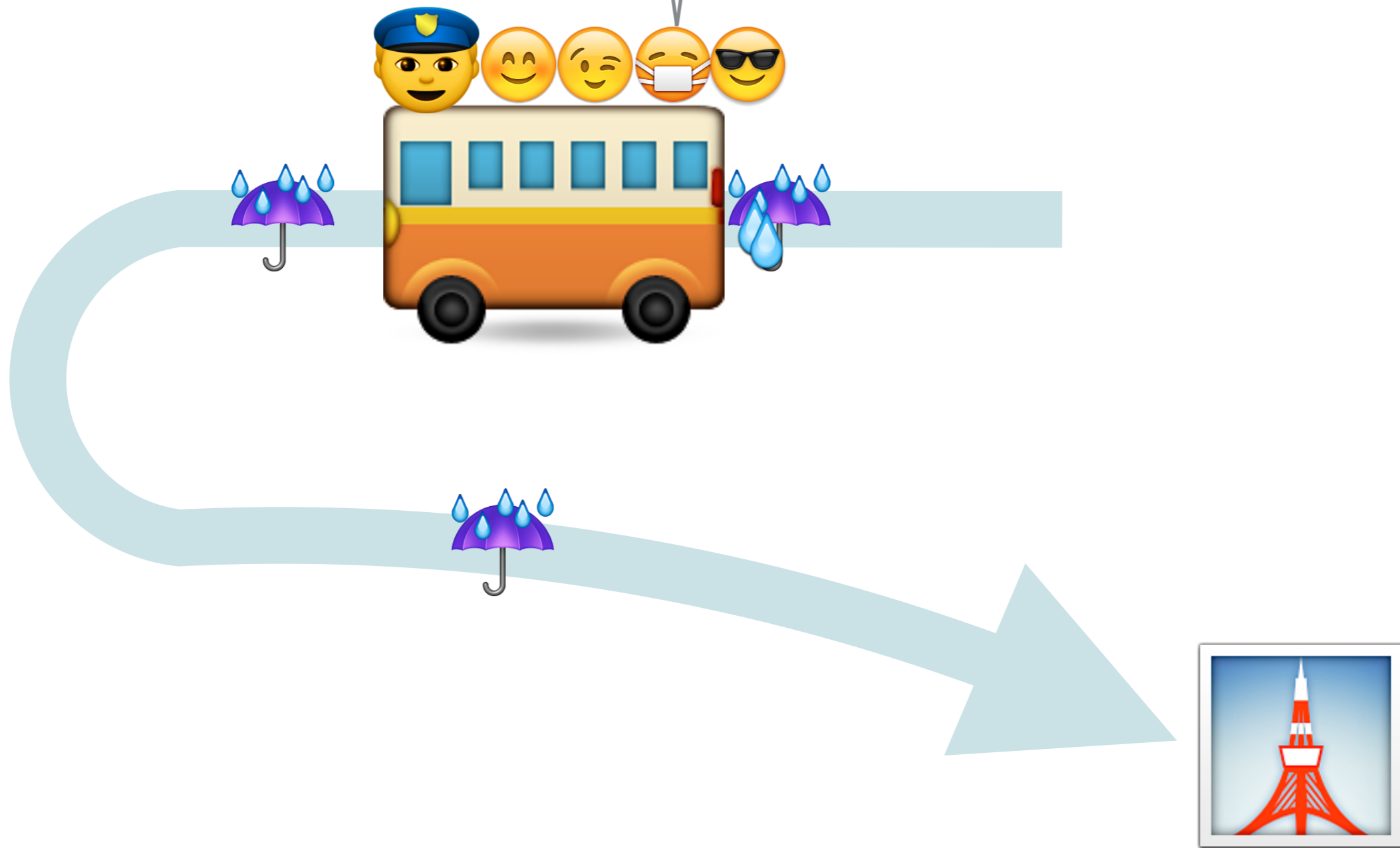
水を飲みます



# 問題概要

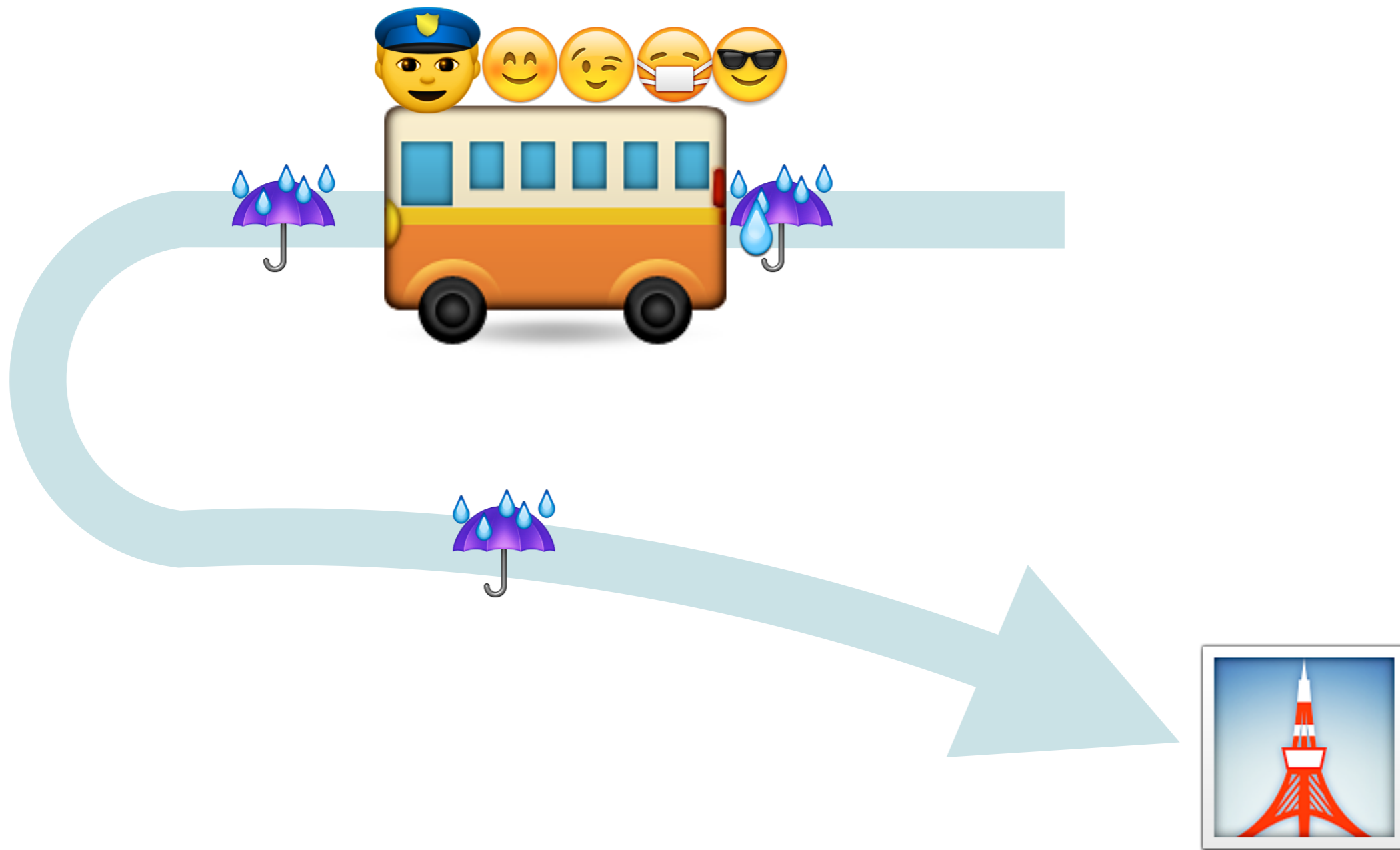
長距離バスが目的地に行きたい

水を飲みます



# 問題概要

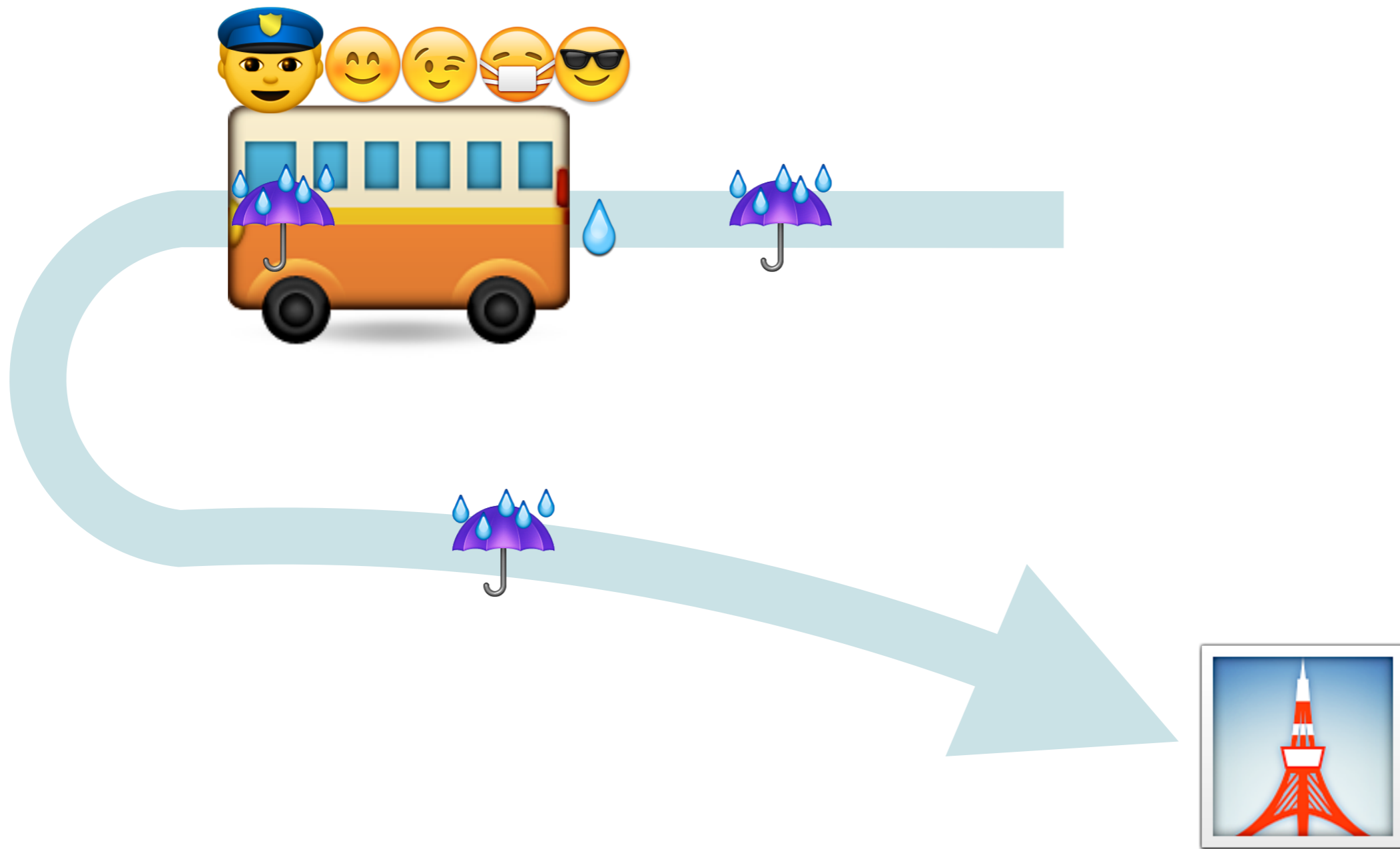
長距離バスが目的地に行きたい





# 問題概要

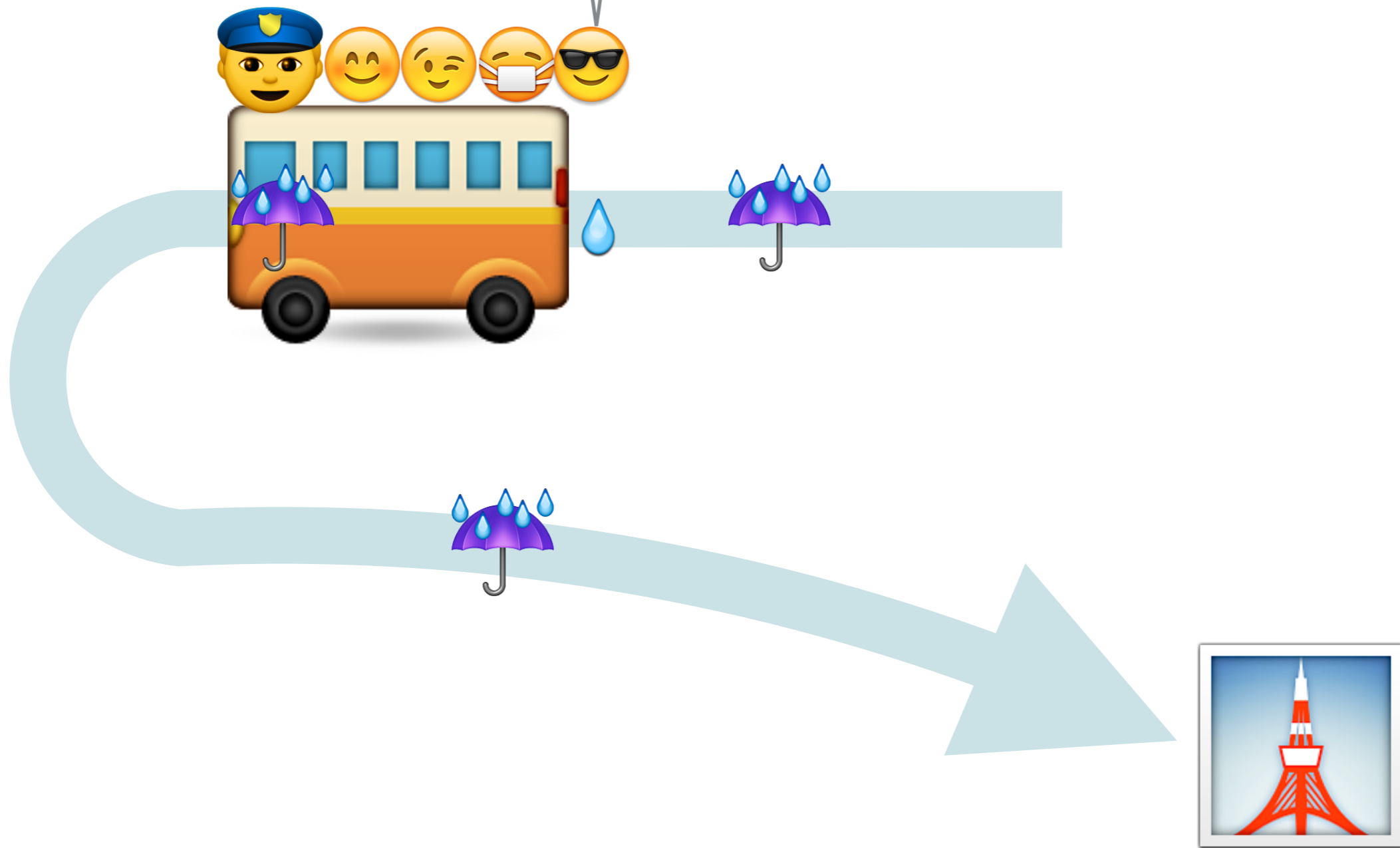
長距離バスが目的地に行きたい



# 問題概要

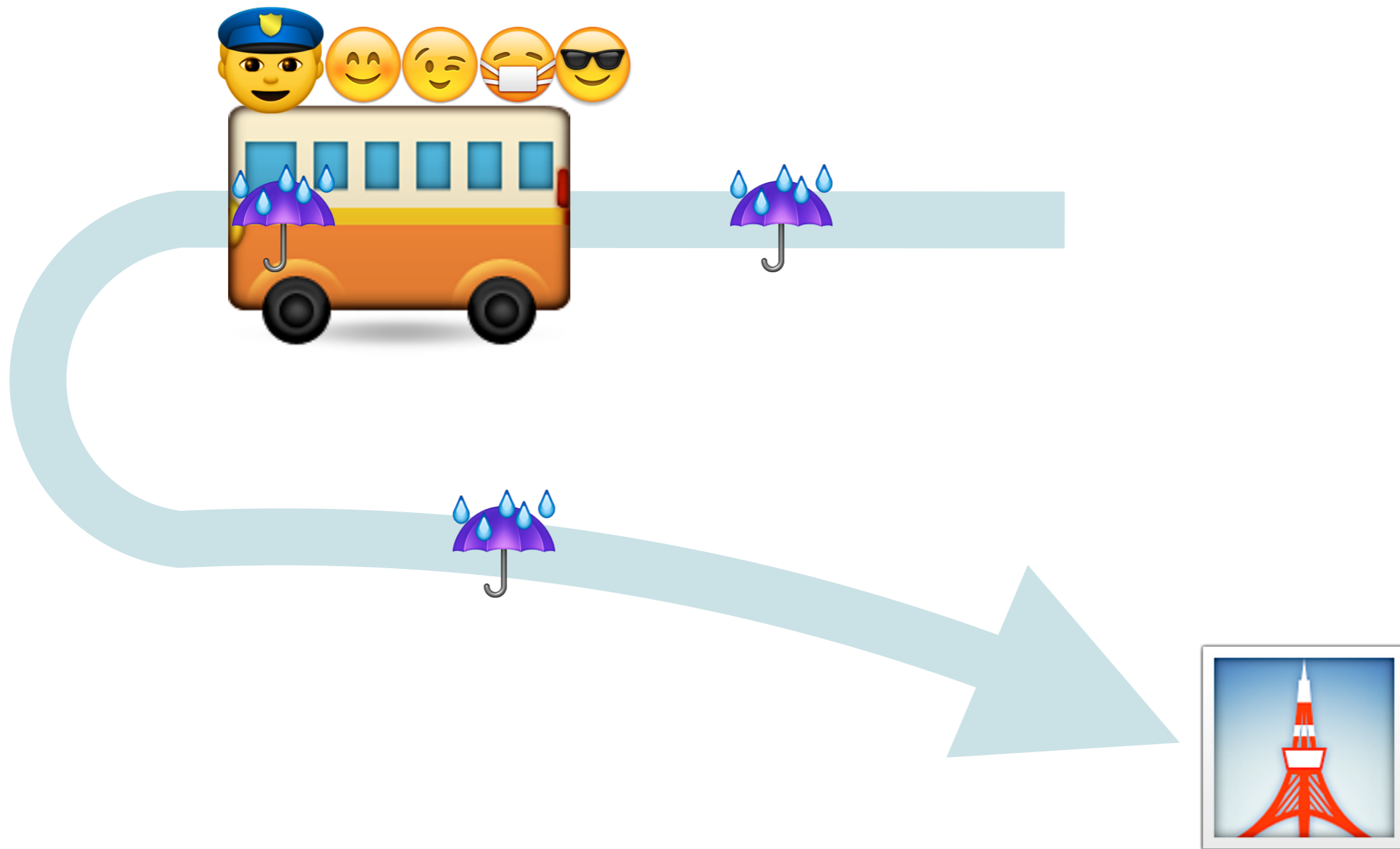
長距離バスが目的地に行きたい

水を飲みます



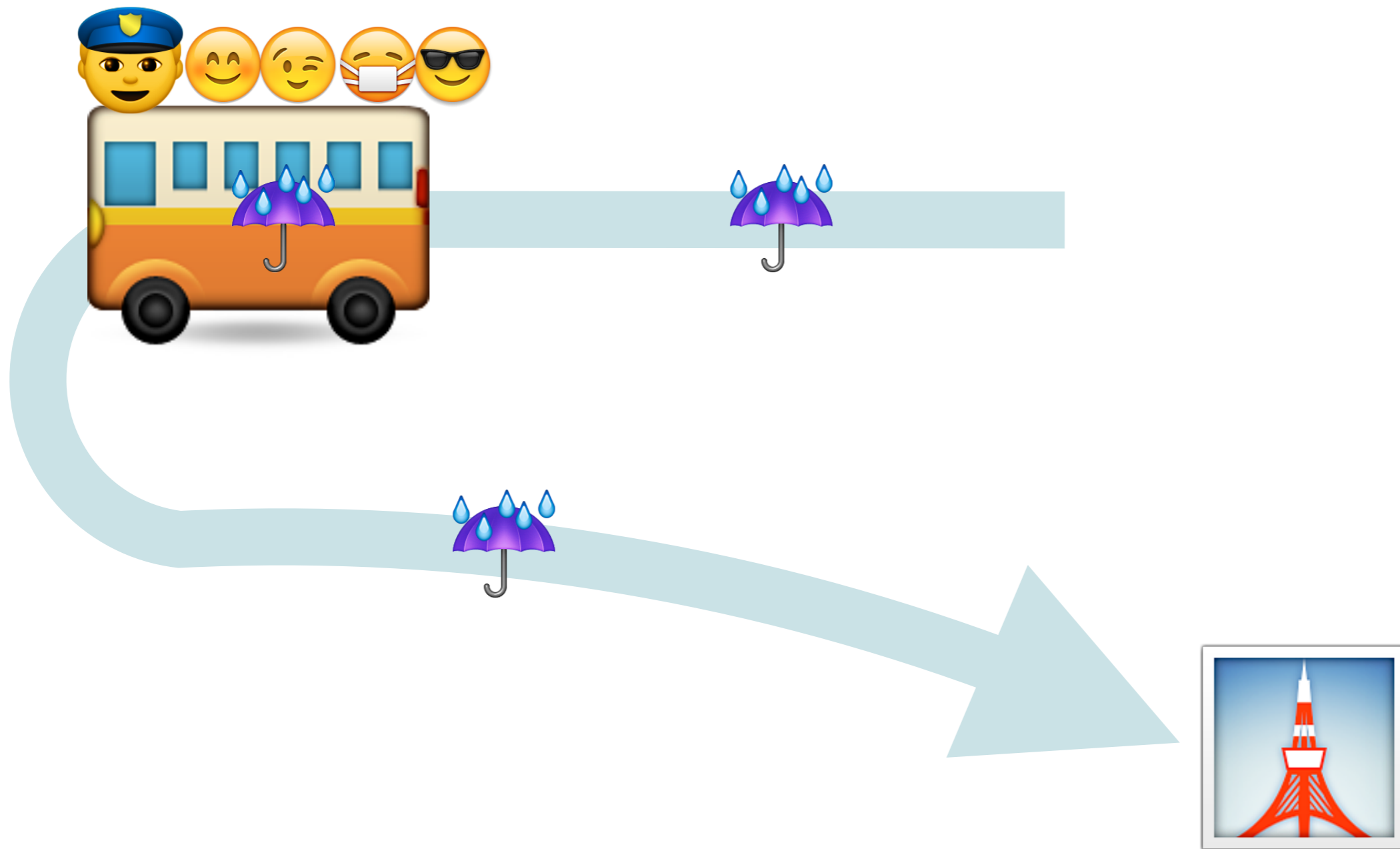
# 問題概要

長距離バスが目的地に行きたい



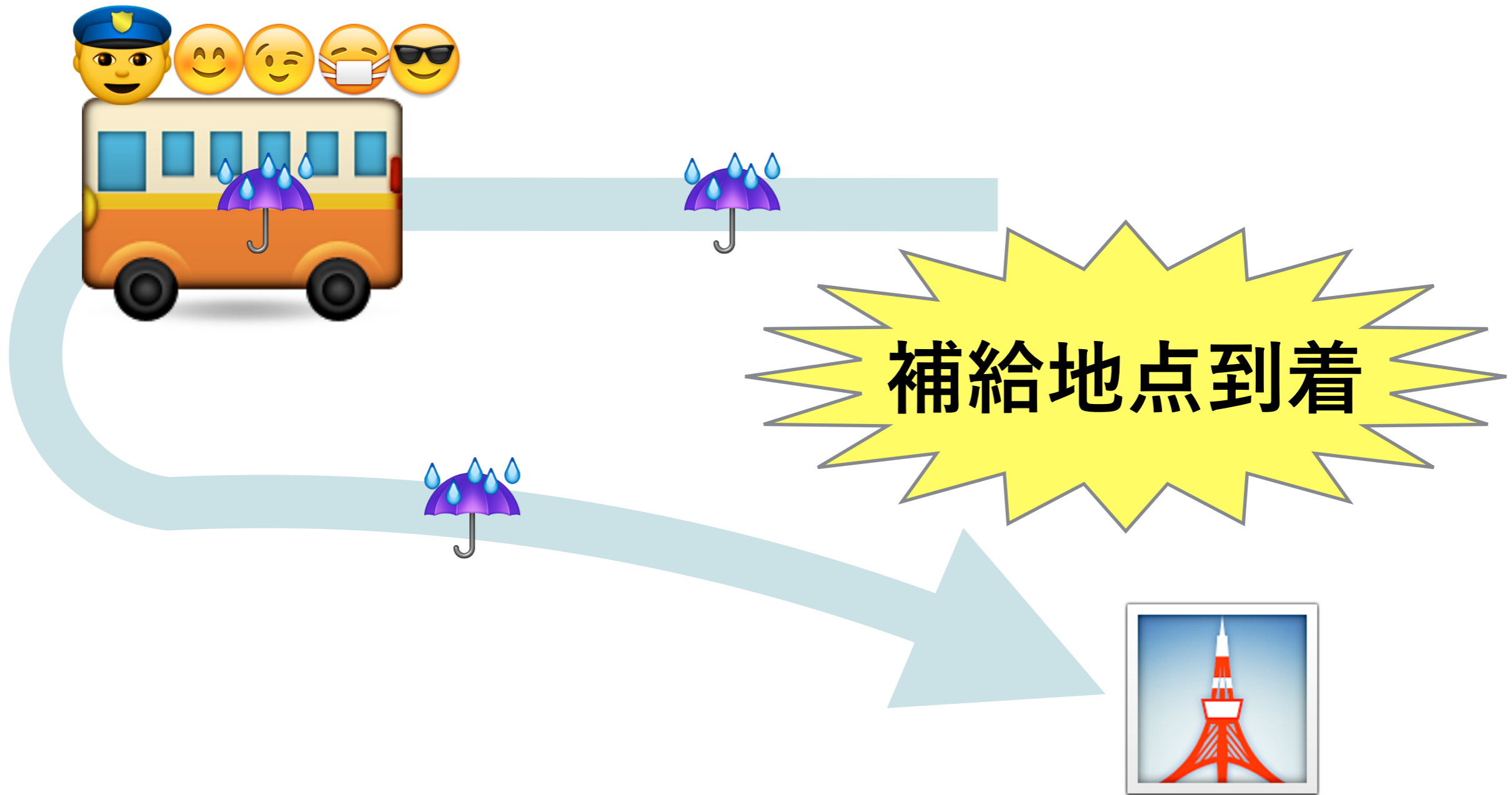
# 問題概要

長距離バスが目的地に行きたい



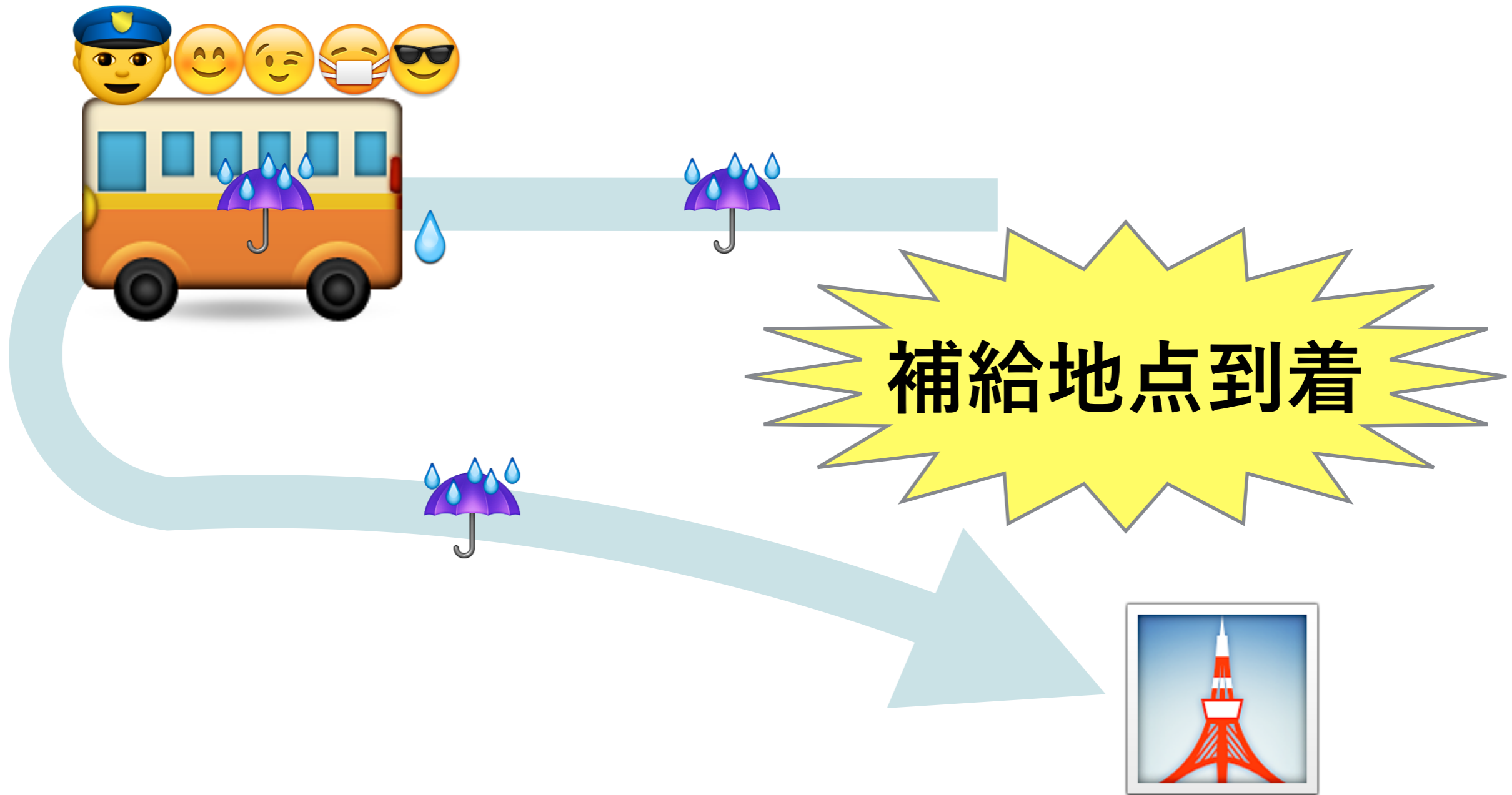
# 問題概要

長距離バスが目的地に行きたい



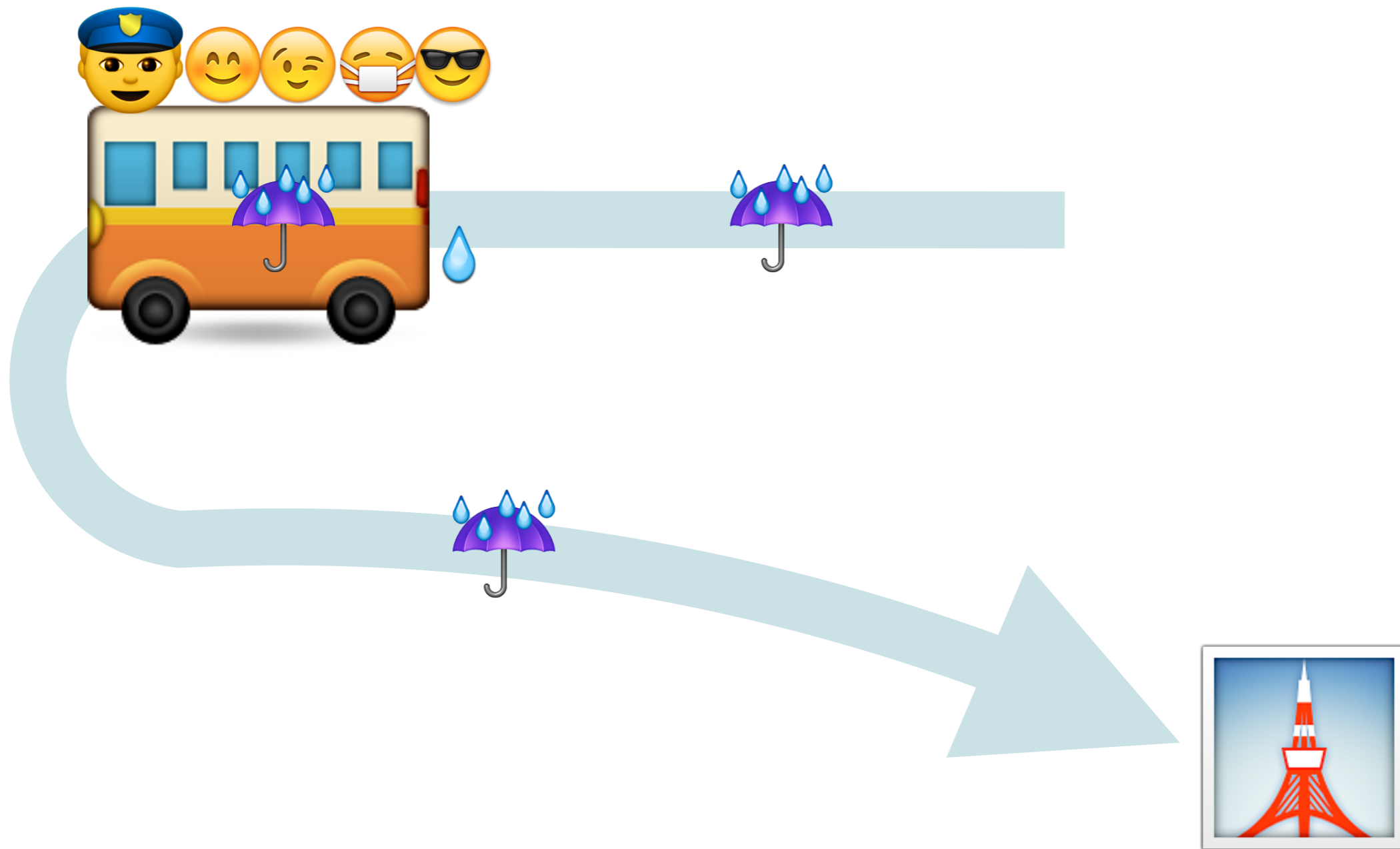
# 問題概要

長距離バスが目的地に行きたい



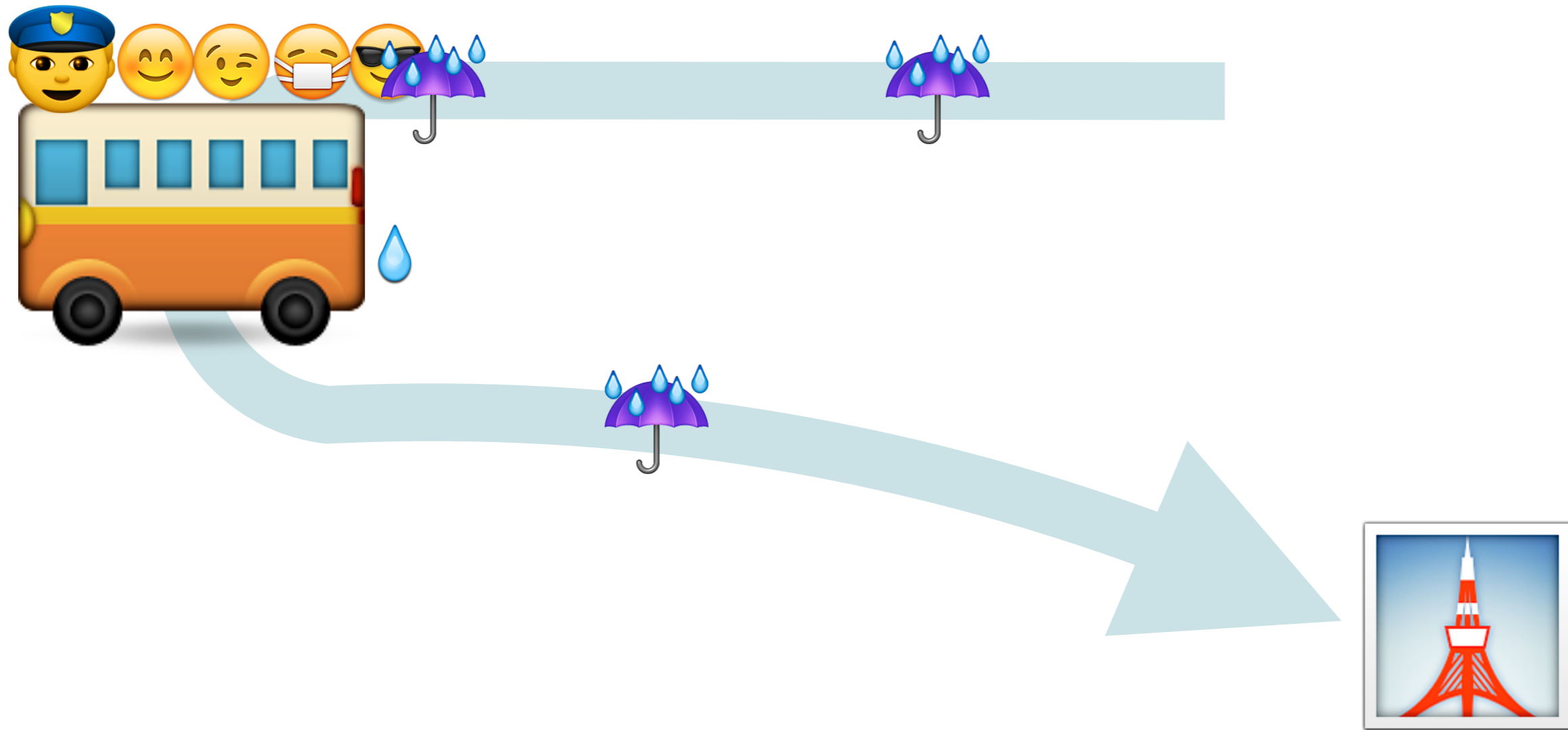
# 問題概要

長距離バスが目的地に行きたい



# 問題概要

長距離バスが目的地に行きたい

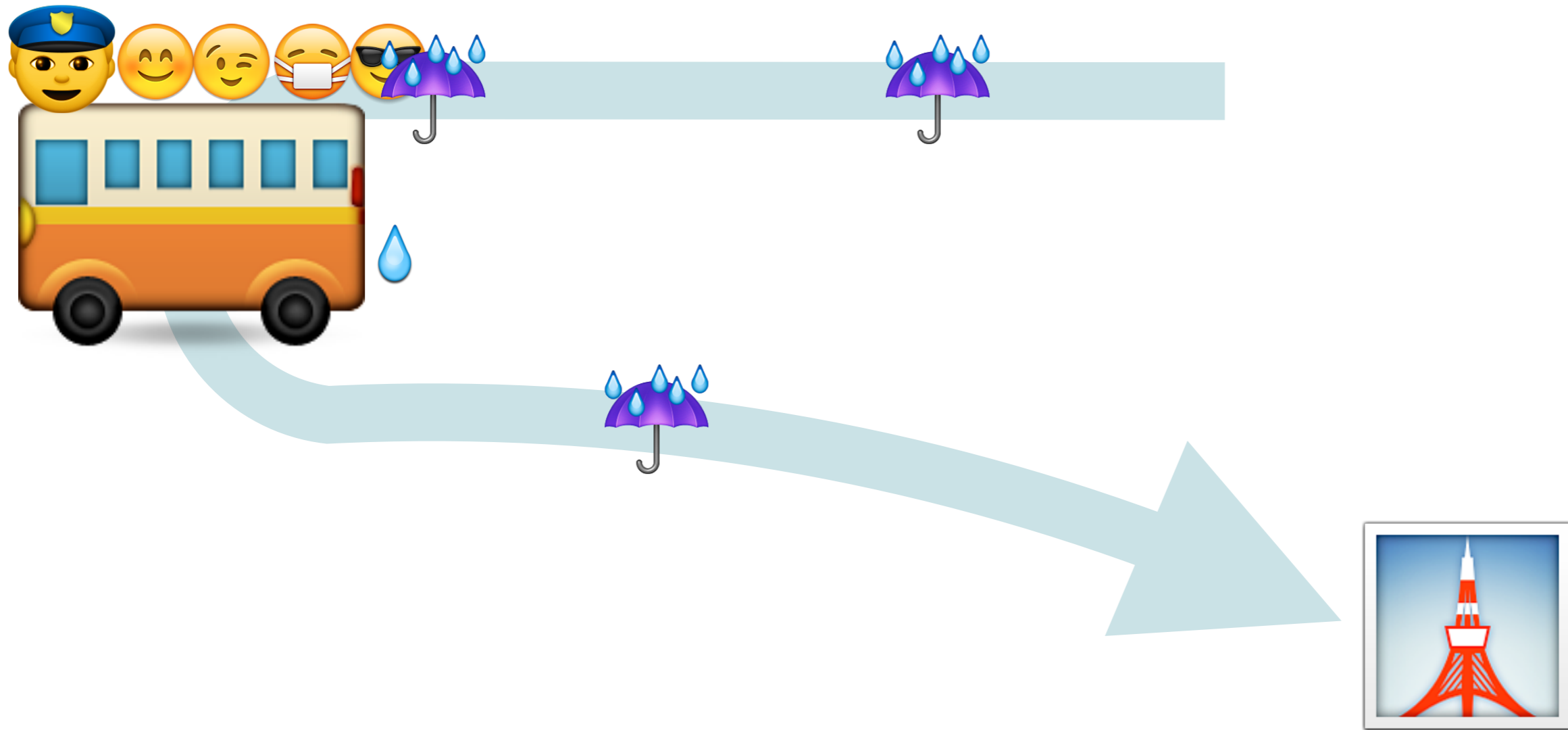




# 問題概要

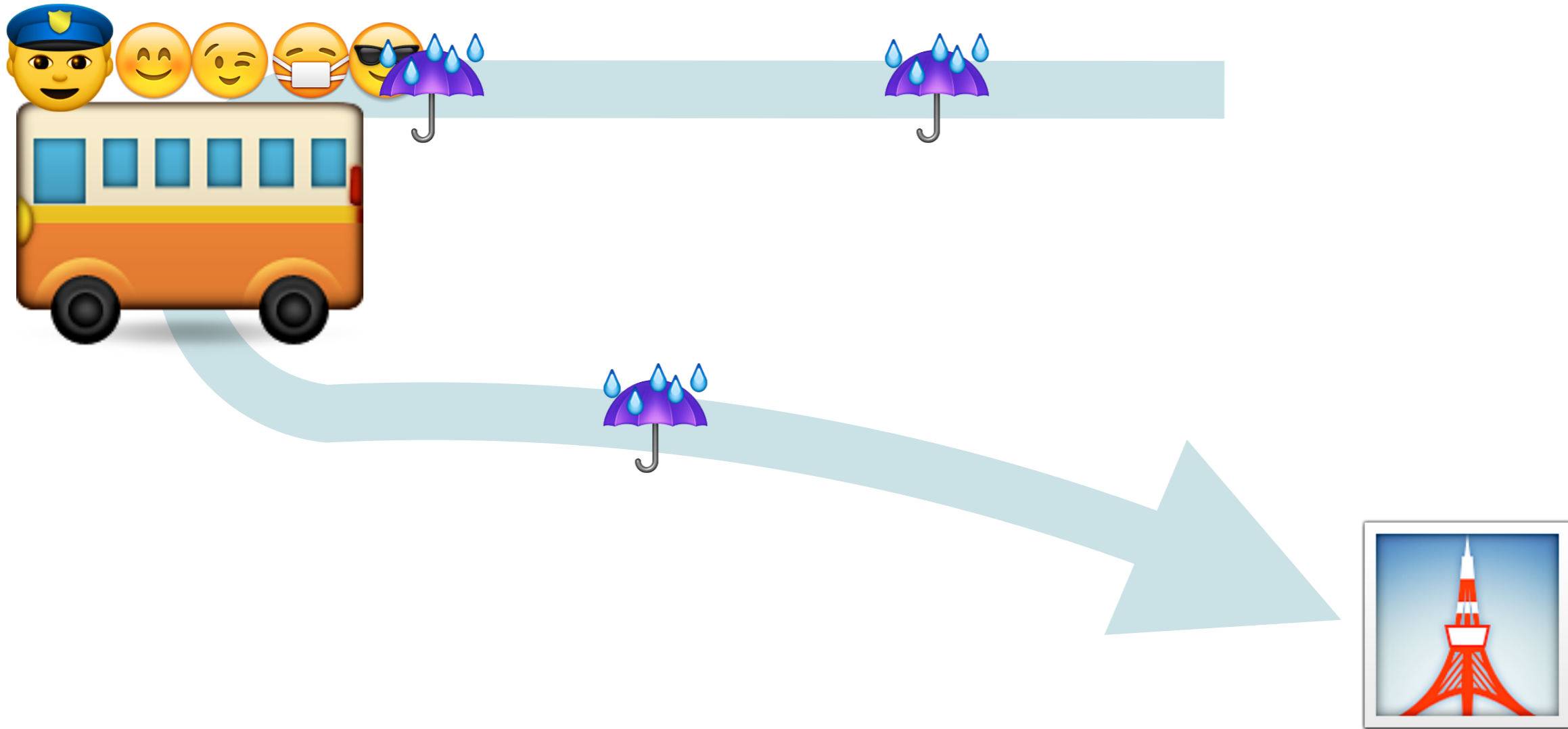
長距離バスが目的地に行きたい

水を飲みます



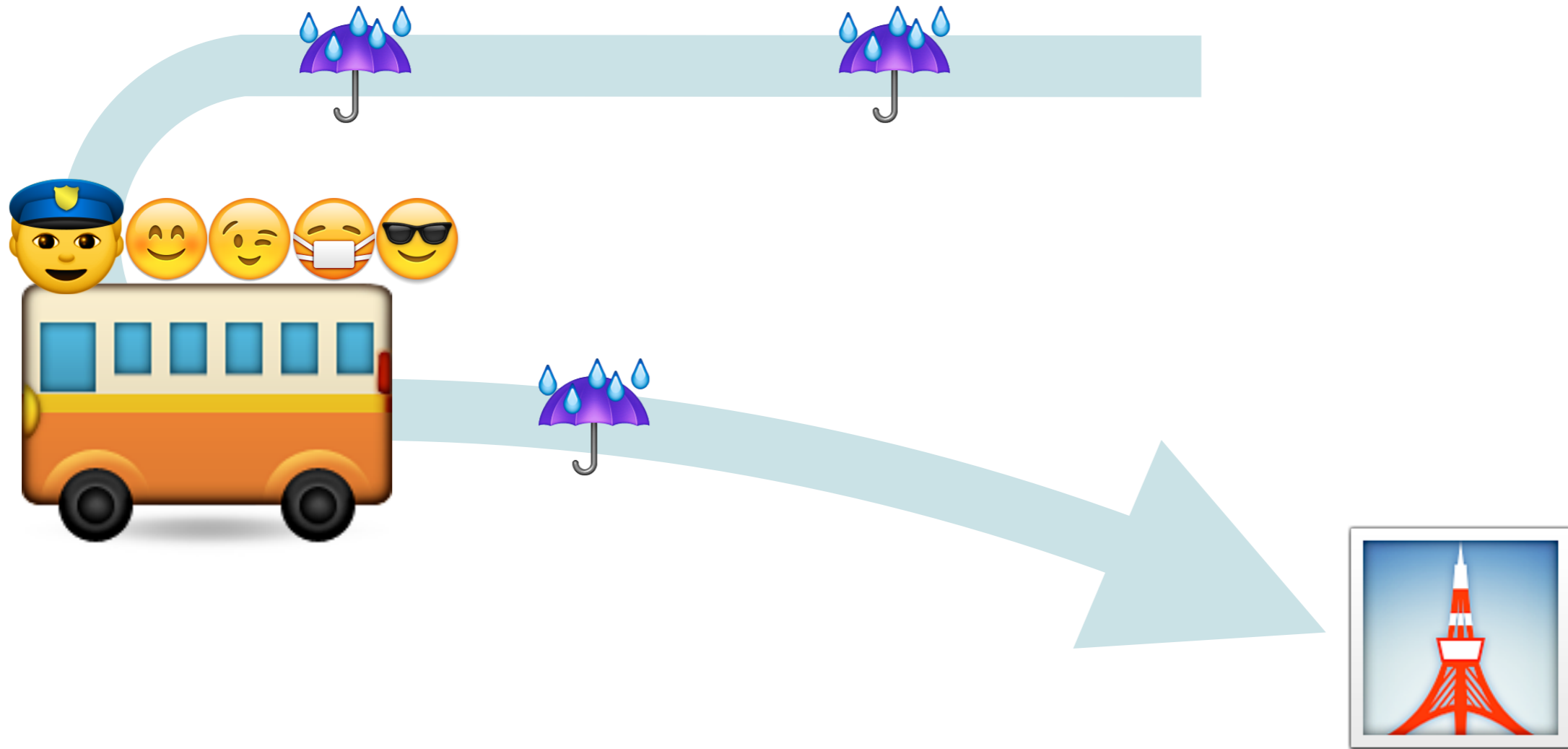
# 問題概要

長距離バスが目的地に行きたい



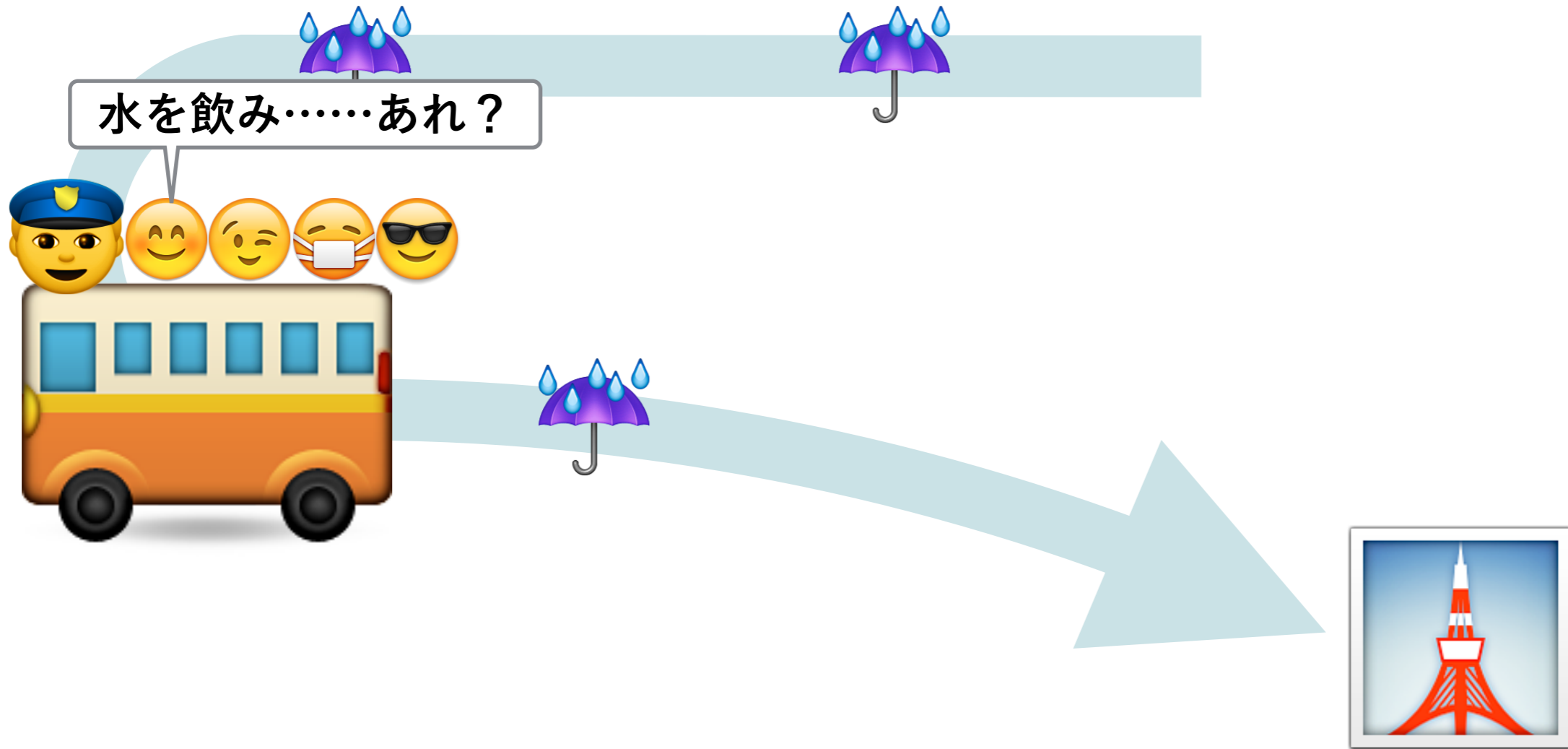
# 問題概要

長距離バスが目的地に行きたい



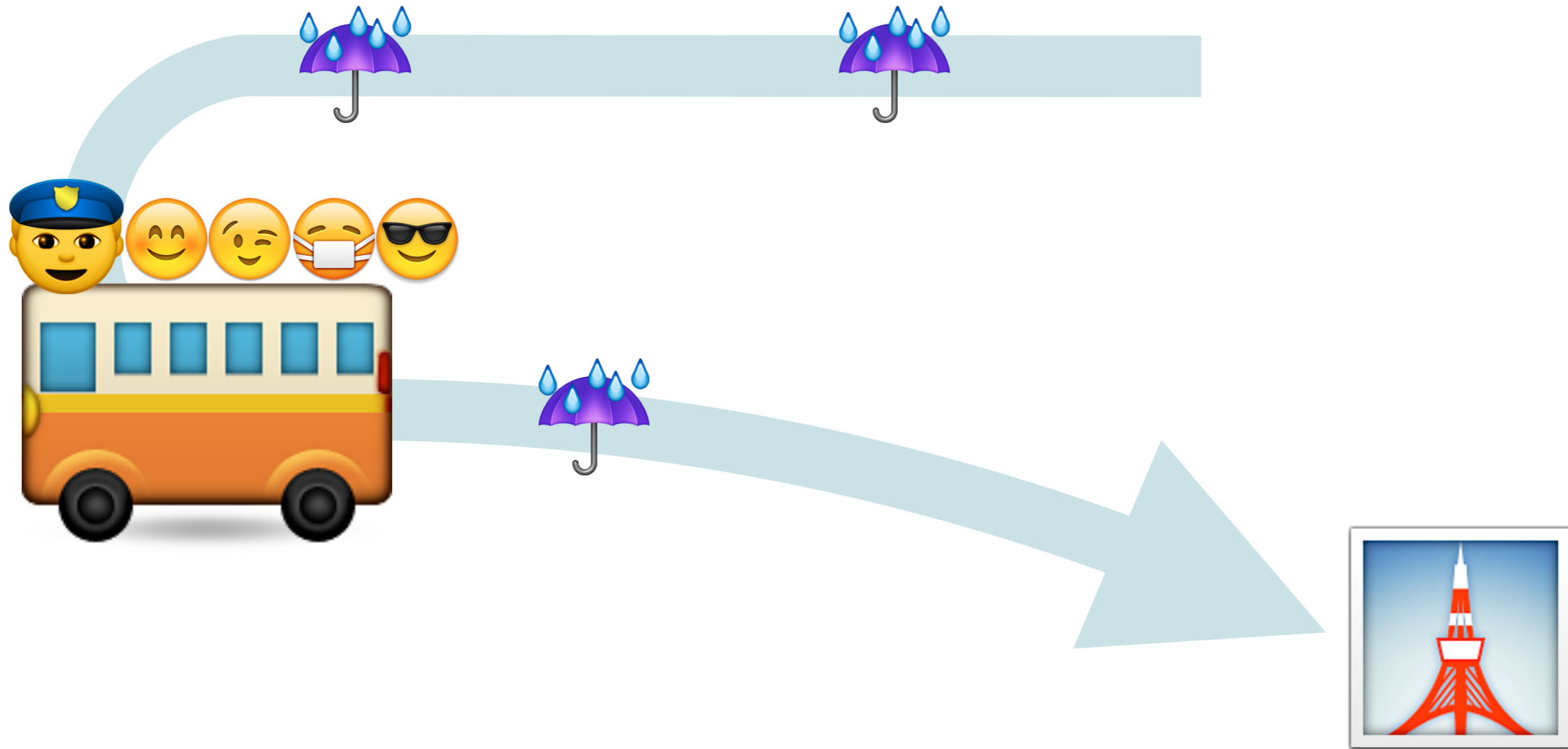
# 問題概要

長距離バスが目的地に行きたい



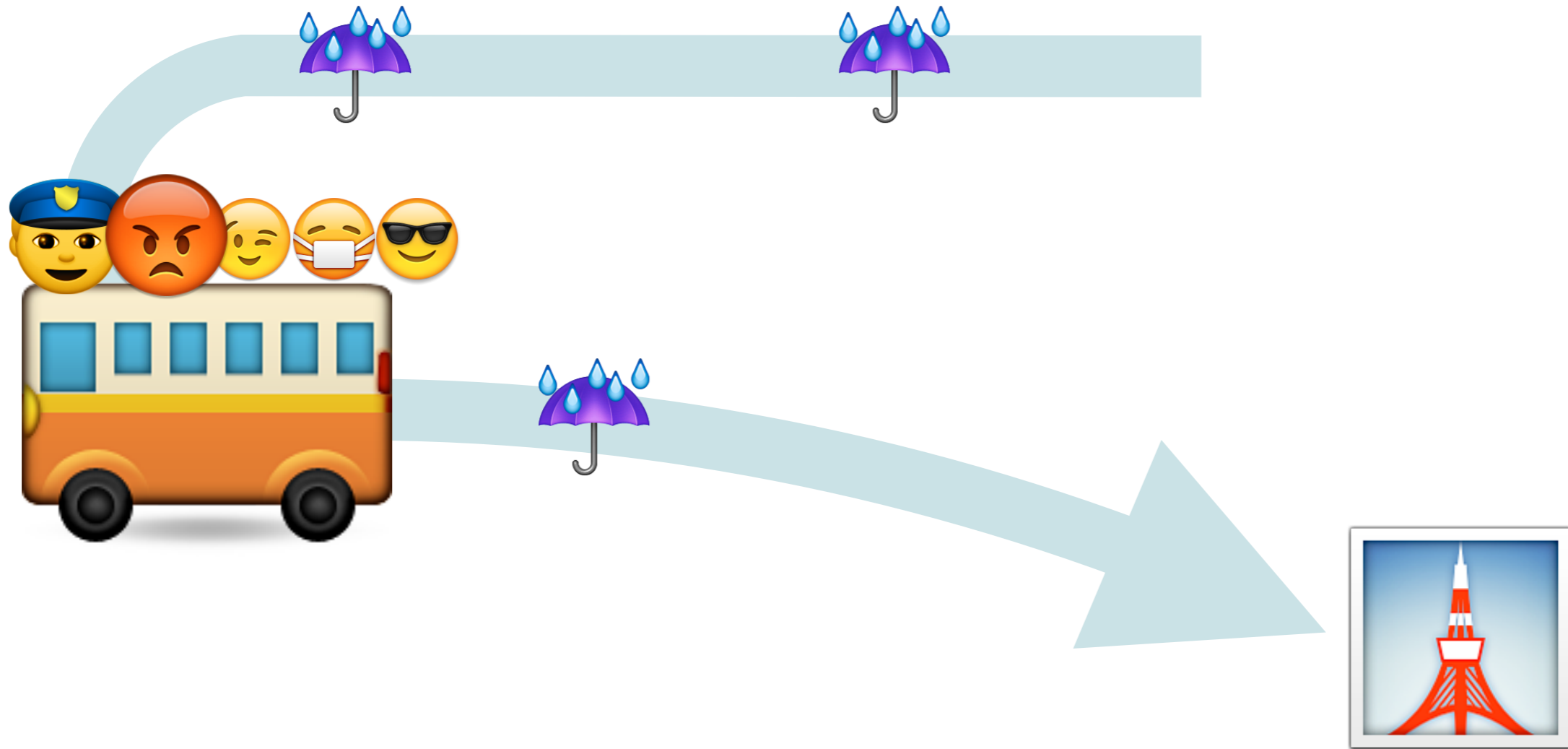
# 問題概要

長距離バスが目的地に行きたい



# 問題概要

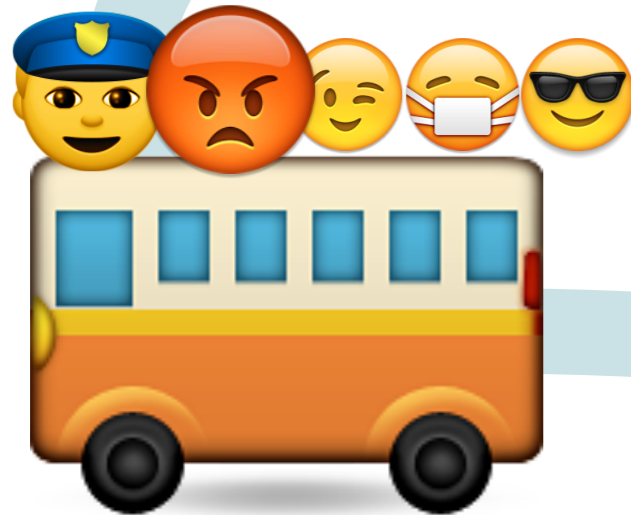
長距離バスが目的地に行きたい



# 問題概要

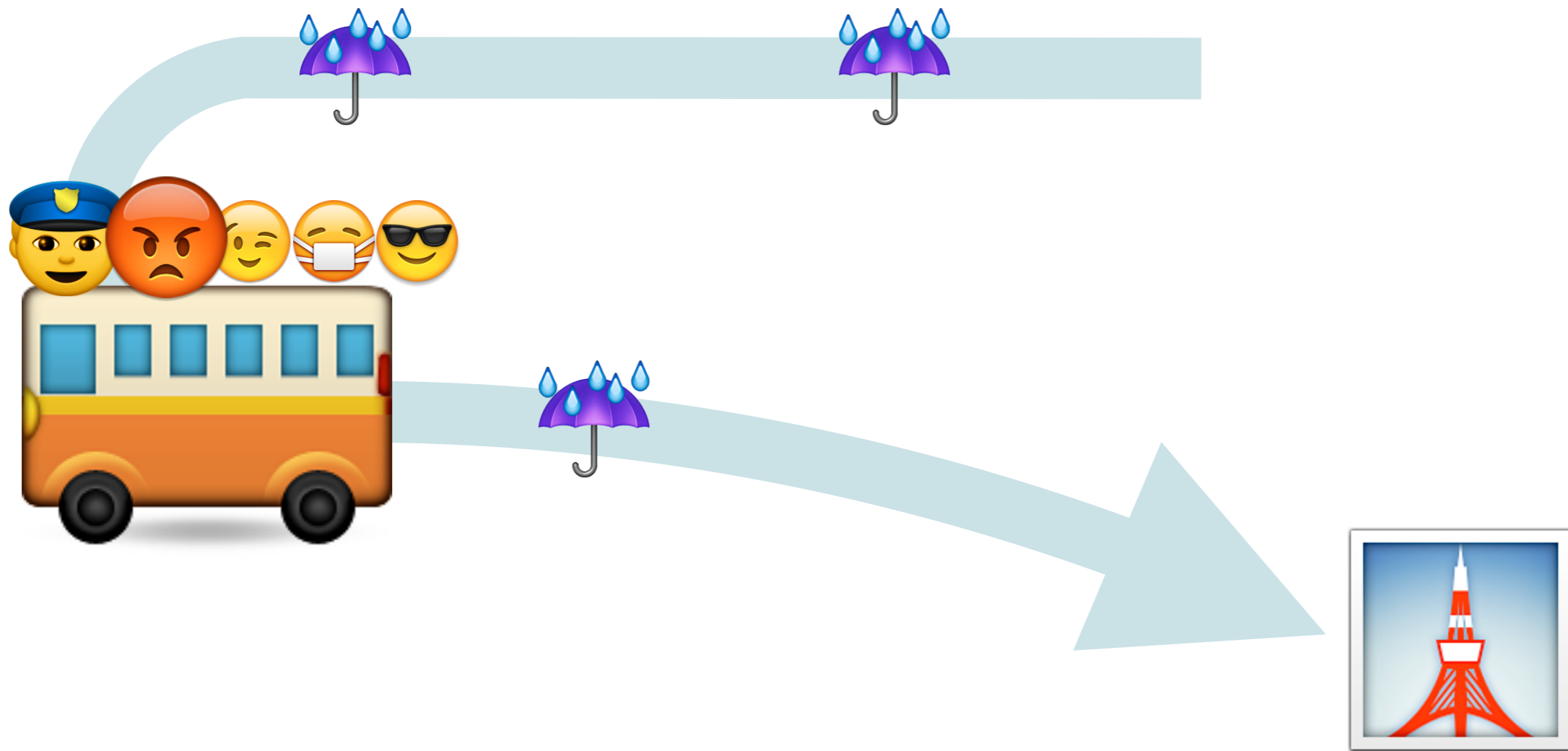
長距離バスが目的地に行きたい

水のないバスに乗っていただけるか！  
帰らせてもらおう!!!



# 問題概要

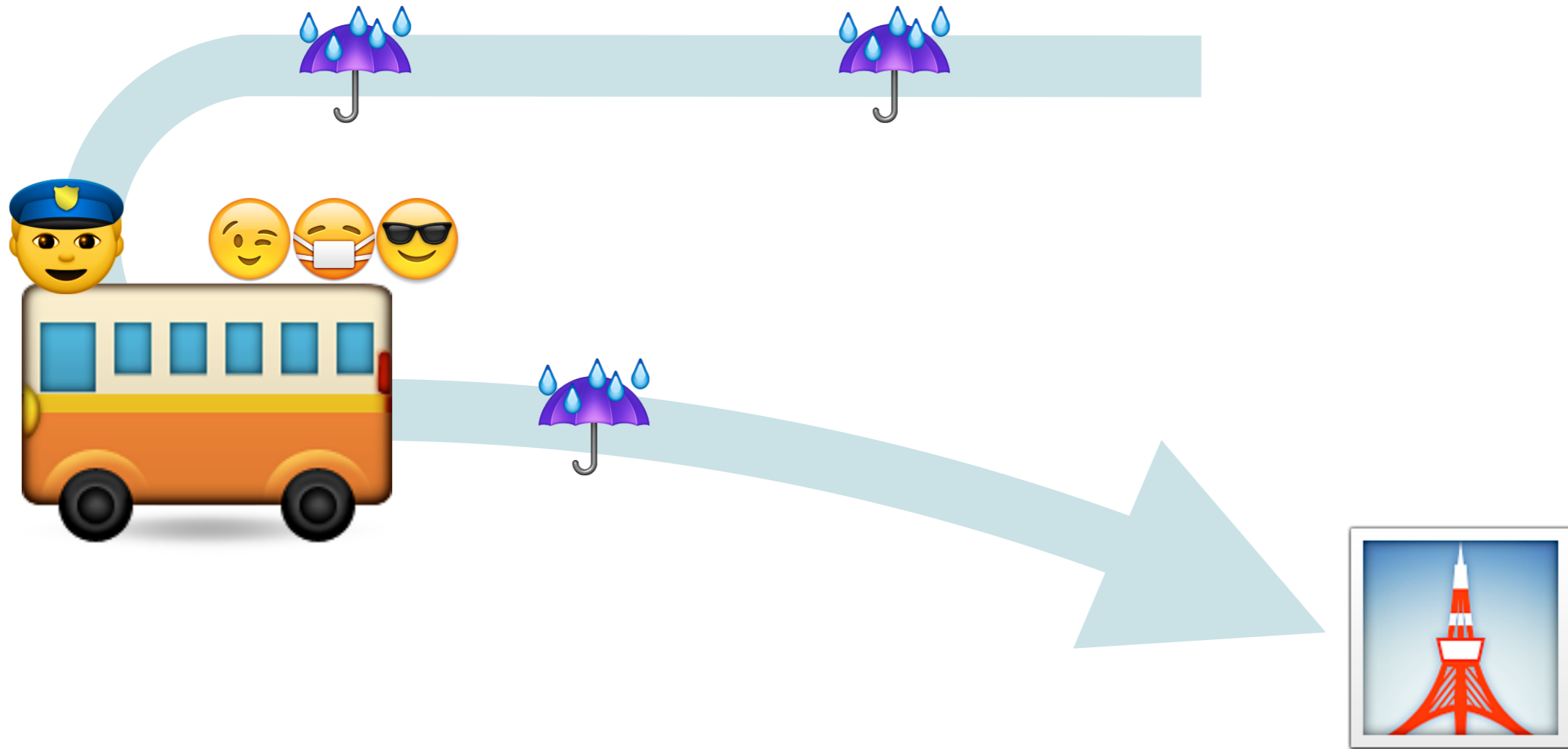
長距離バスが目的地に行きたい





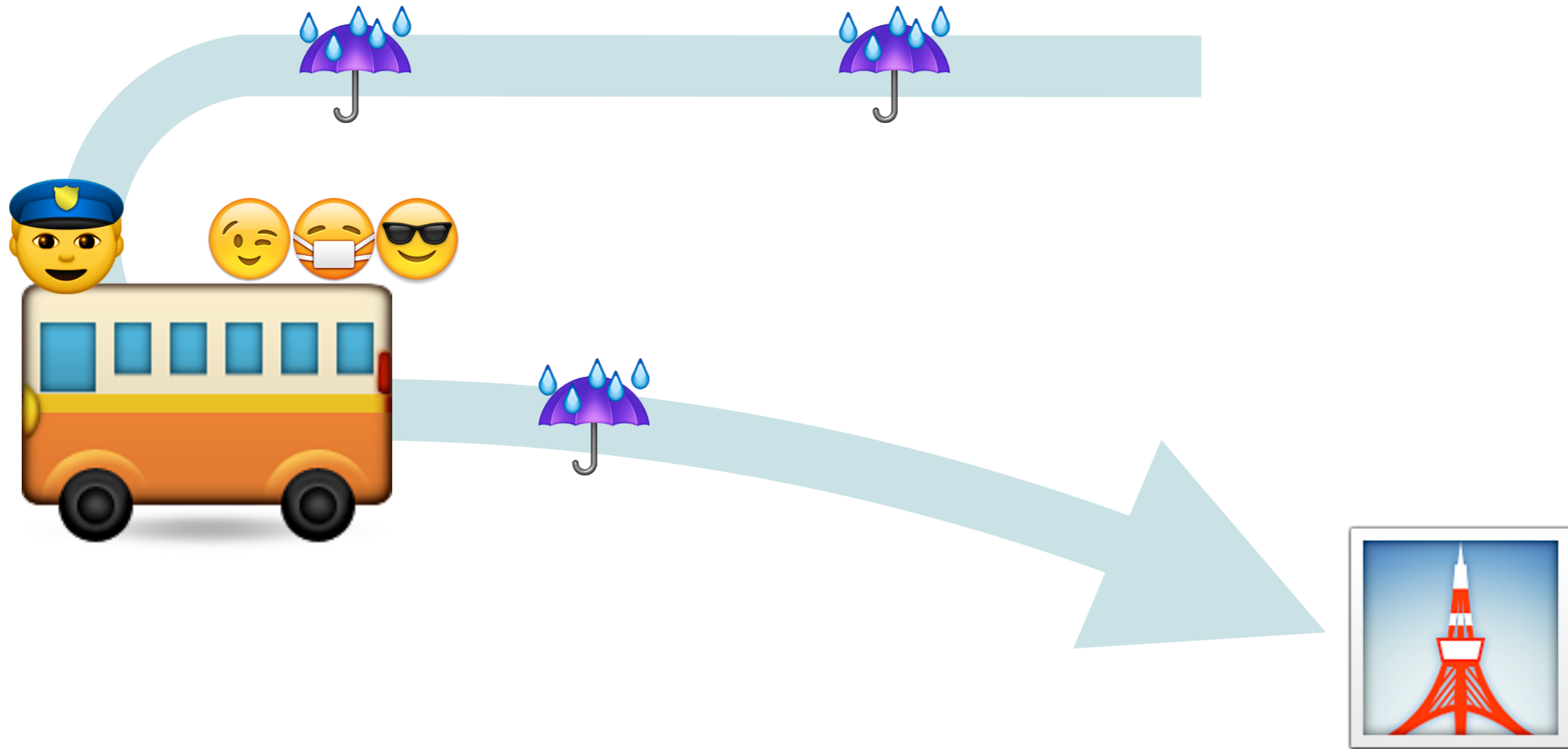
# 問題概要

長距離バスが目的地に行きたい



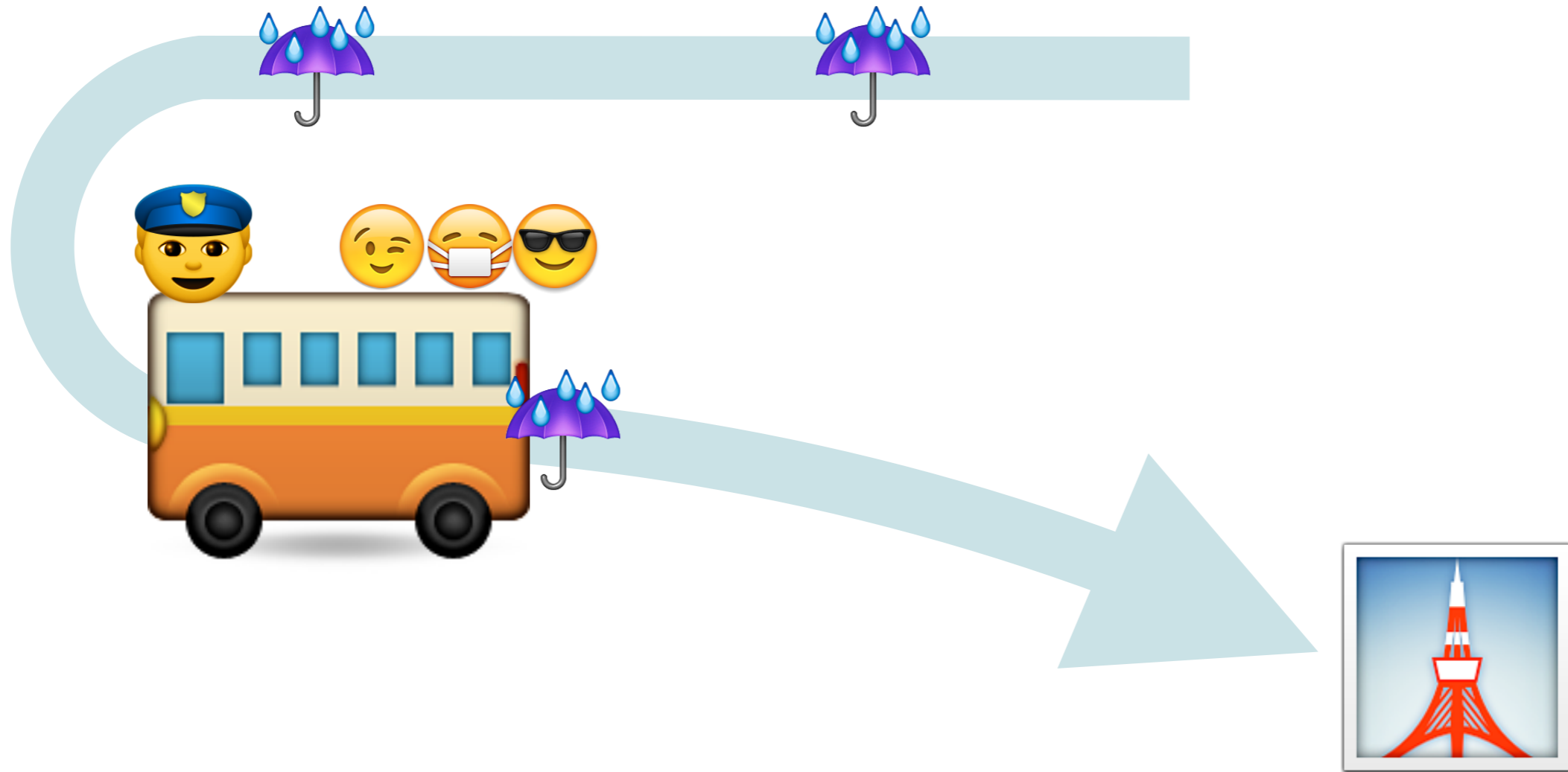
# 問題概要

長距離バスが目的地に行きたい



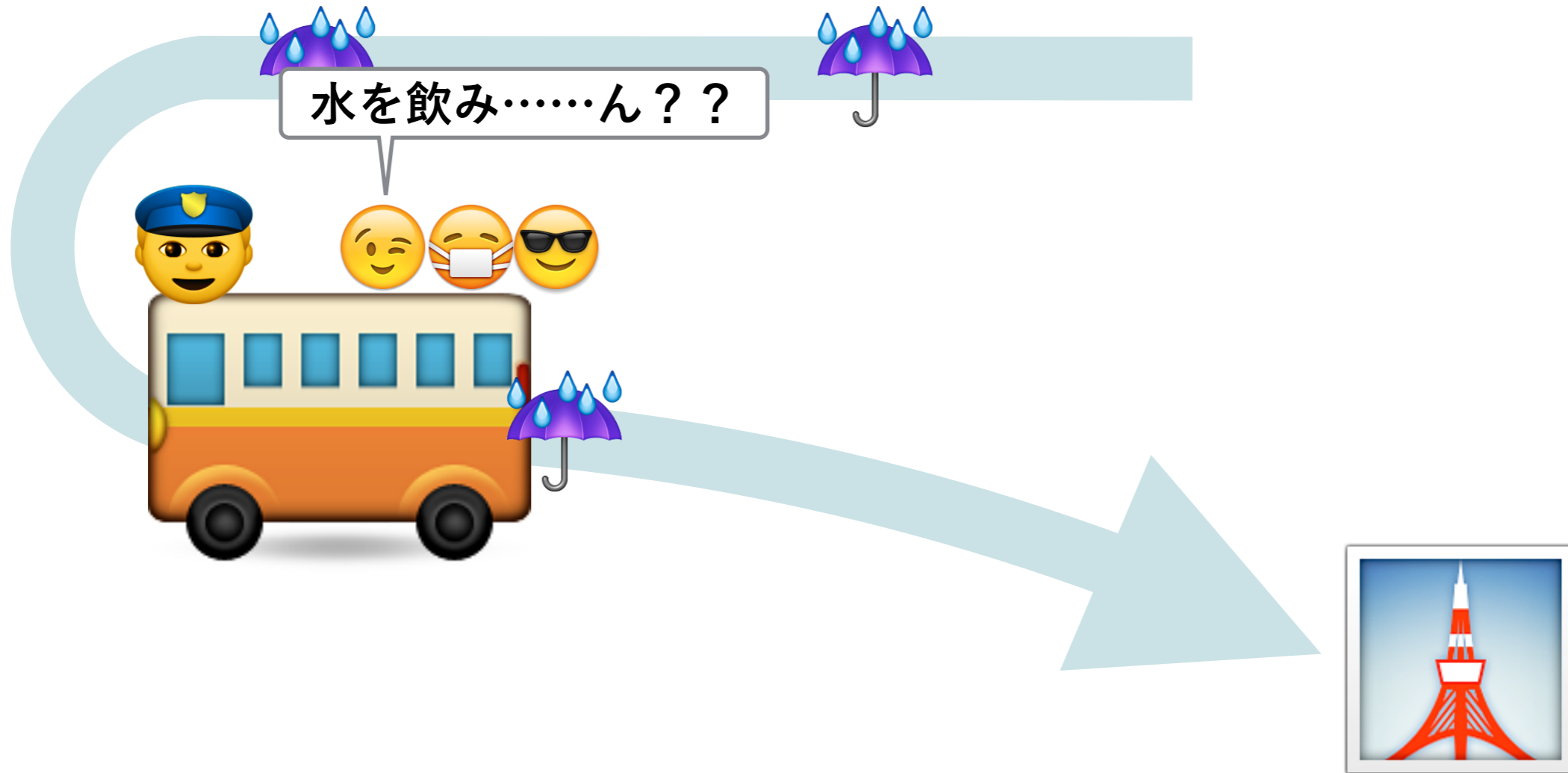
# 問題概要

長距離バスが目的地に行きたい



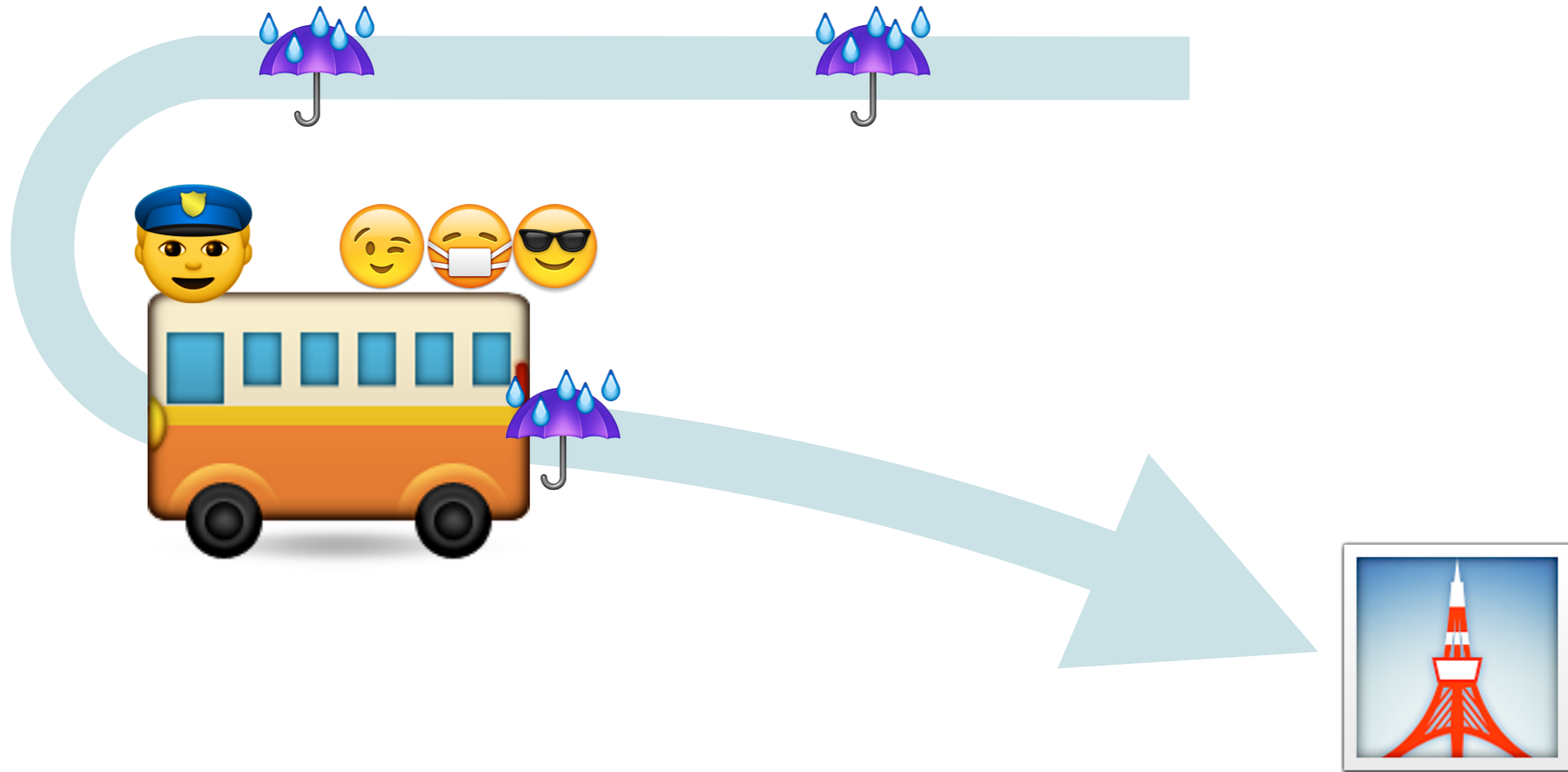
# 問題概要

長距離バスが目的地に行きたい



# 問題概要

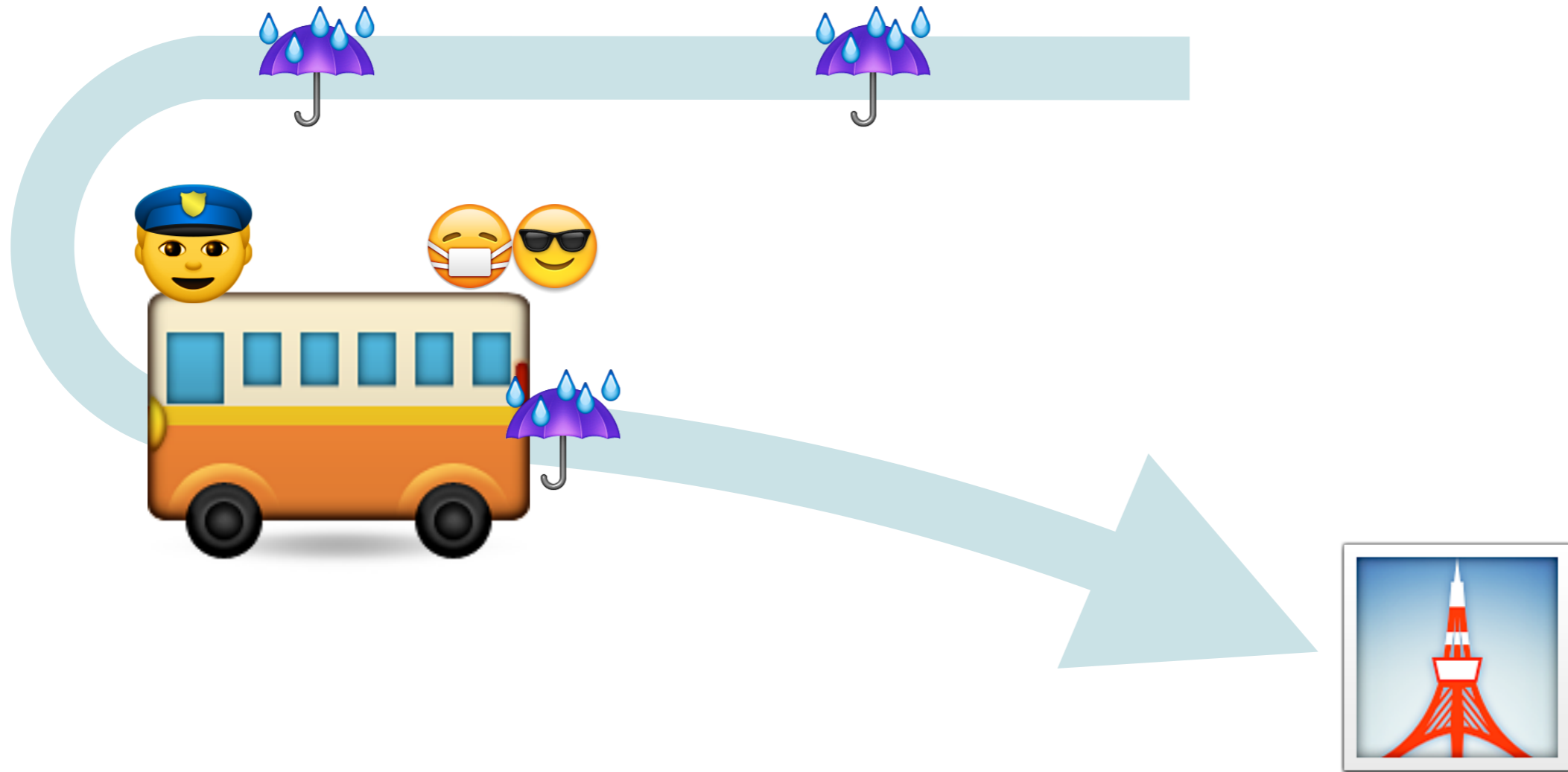
長距離バスが目的地に行きたい





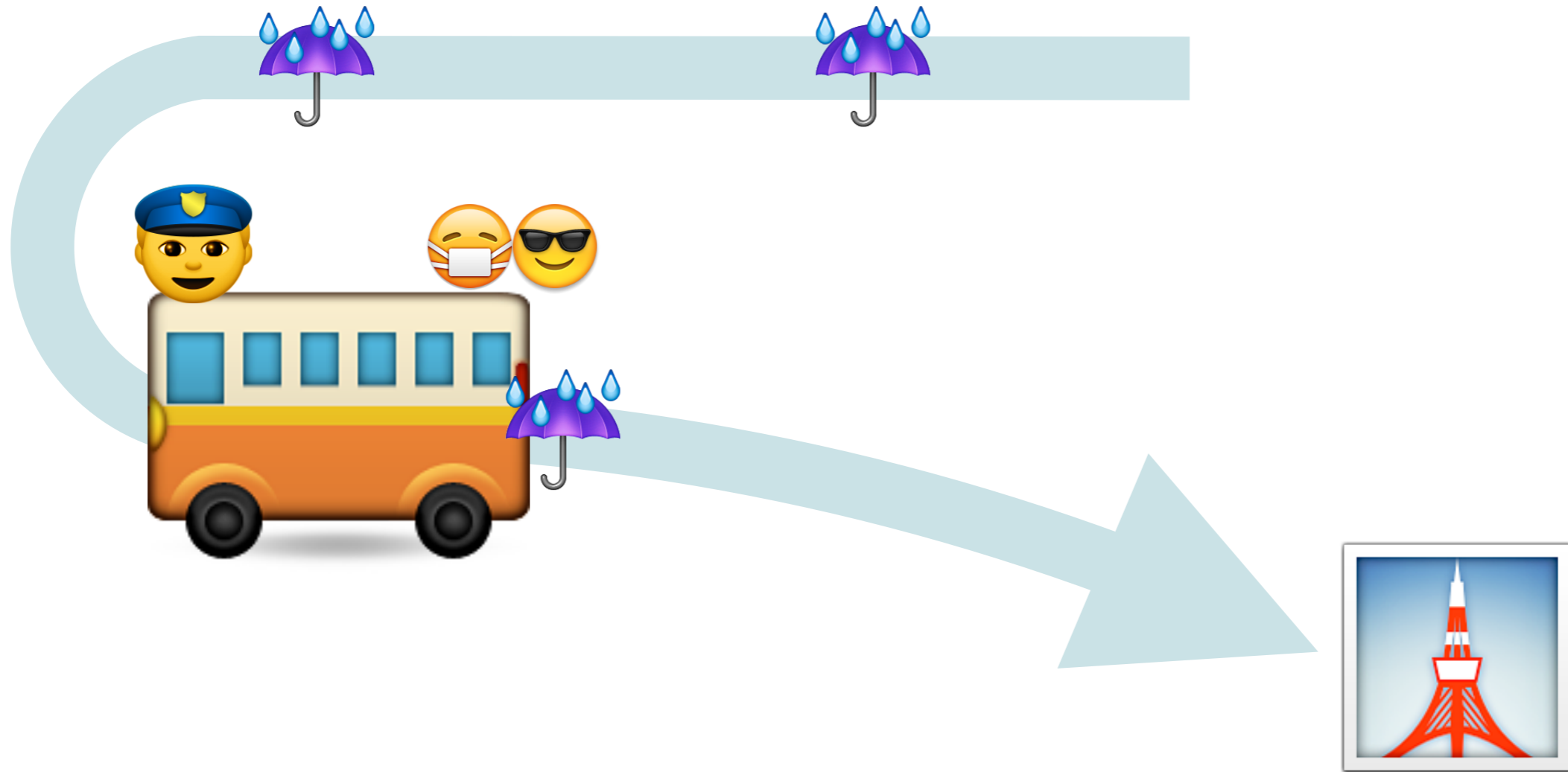
# 問題概要

長距離バスが目的地に行きたい



# 問題概要

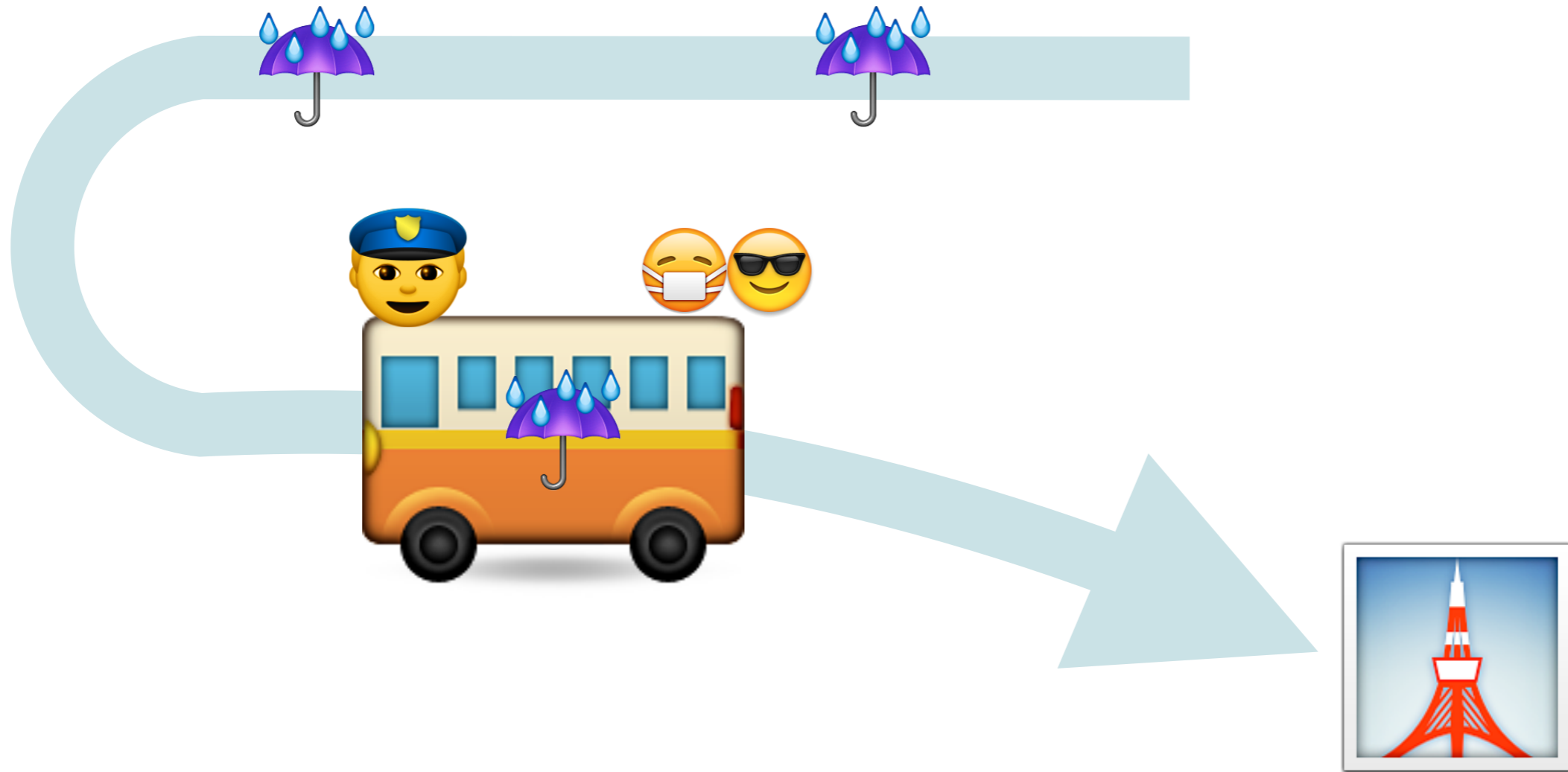
長距離バスが目的地に行きたい





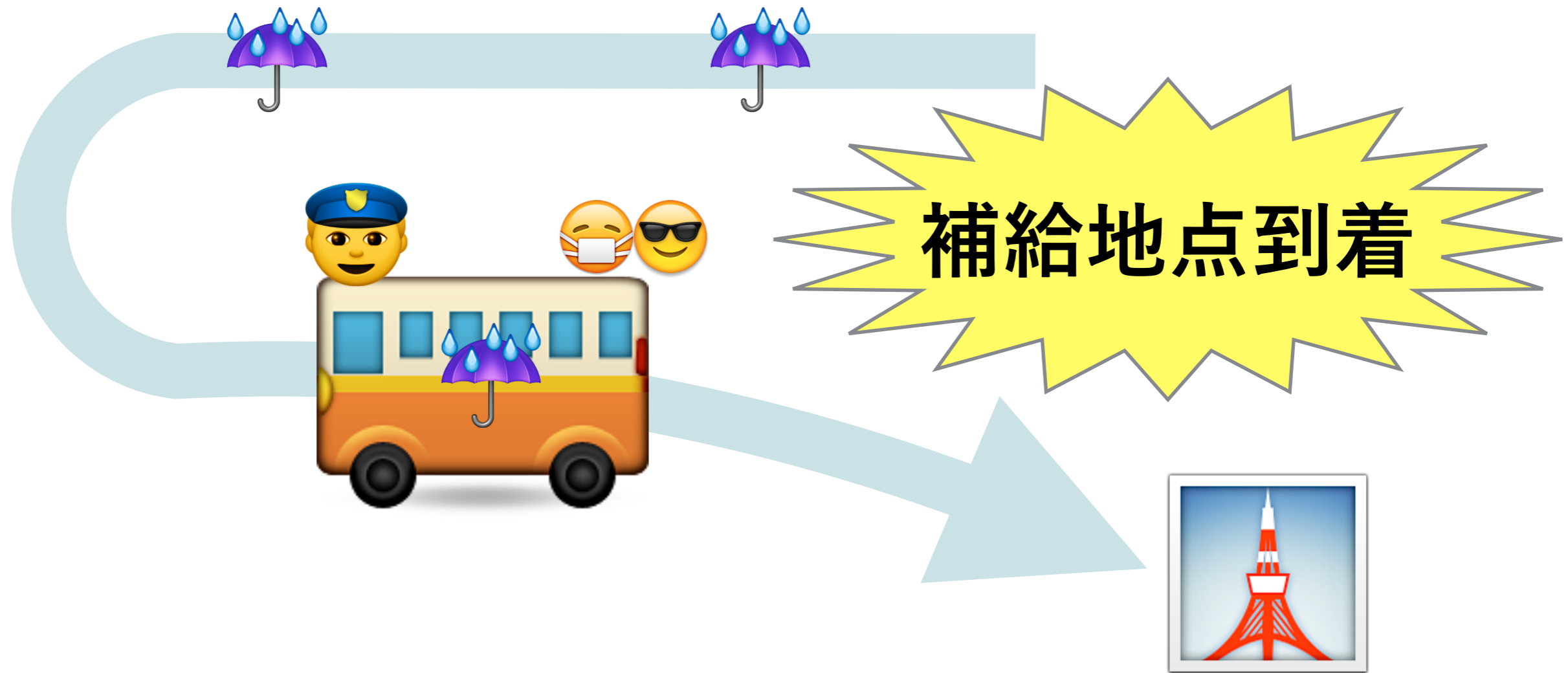
# 問題概要

長距離バスが目的地に行きたい



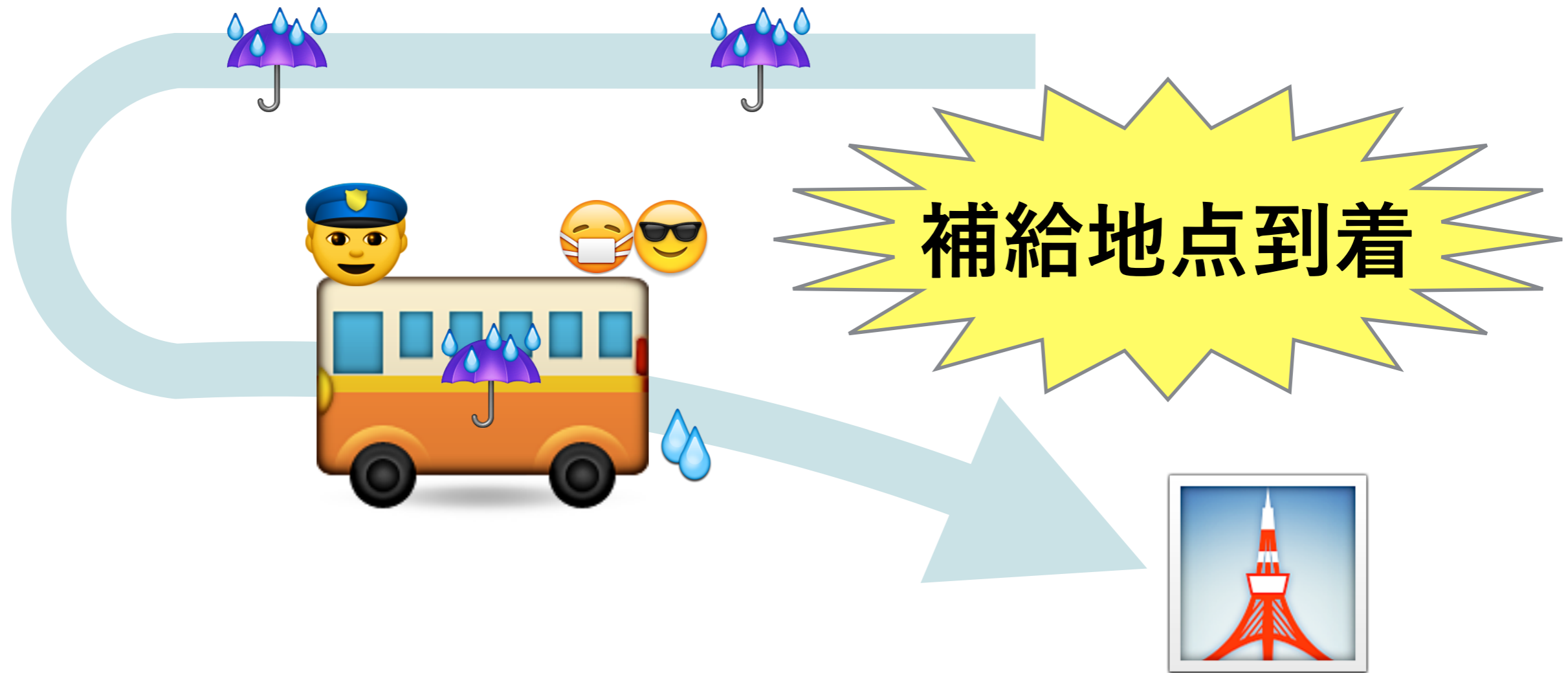
# 問題概要

長距離バスが目的地に行きたい



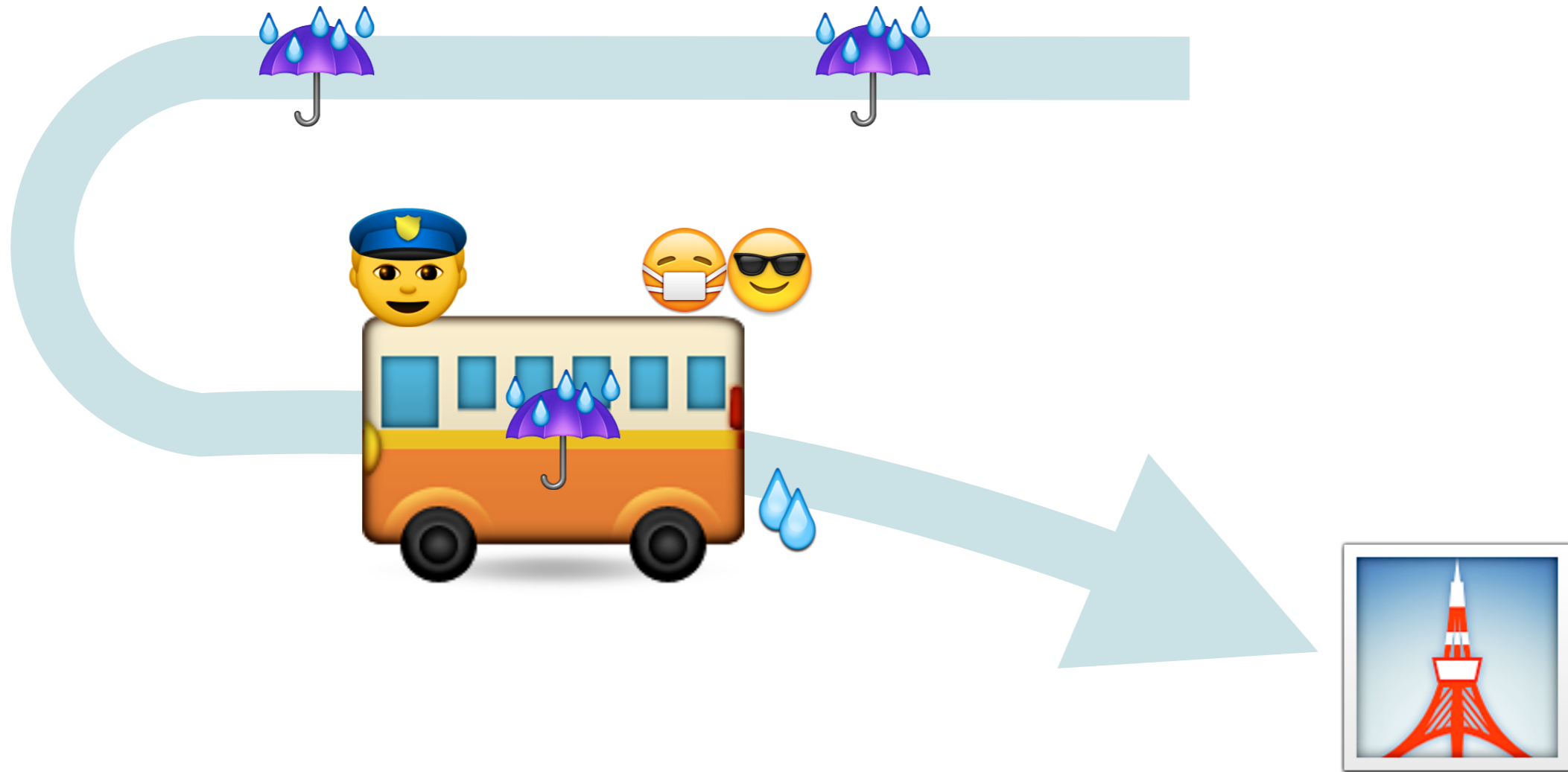
# 問題概要

長距離バスが目的地に行きたい



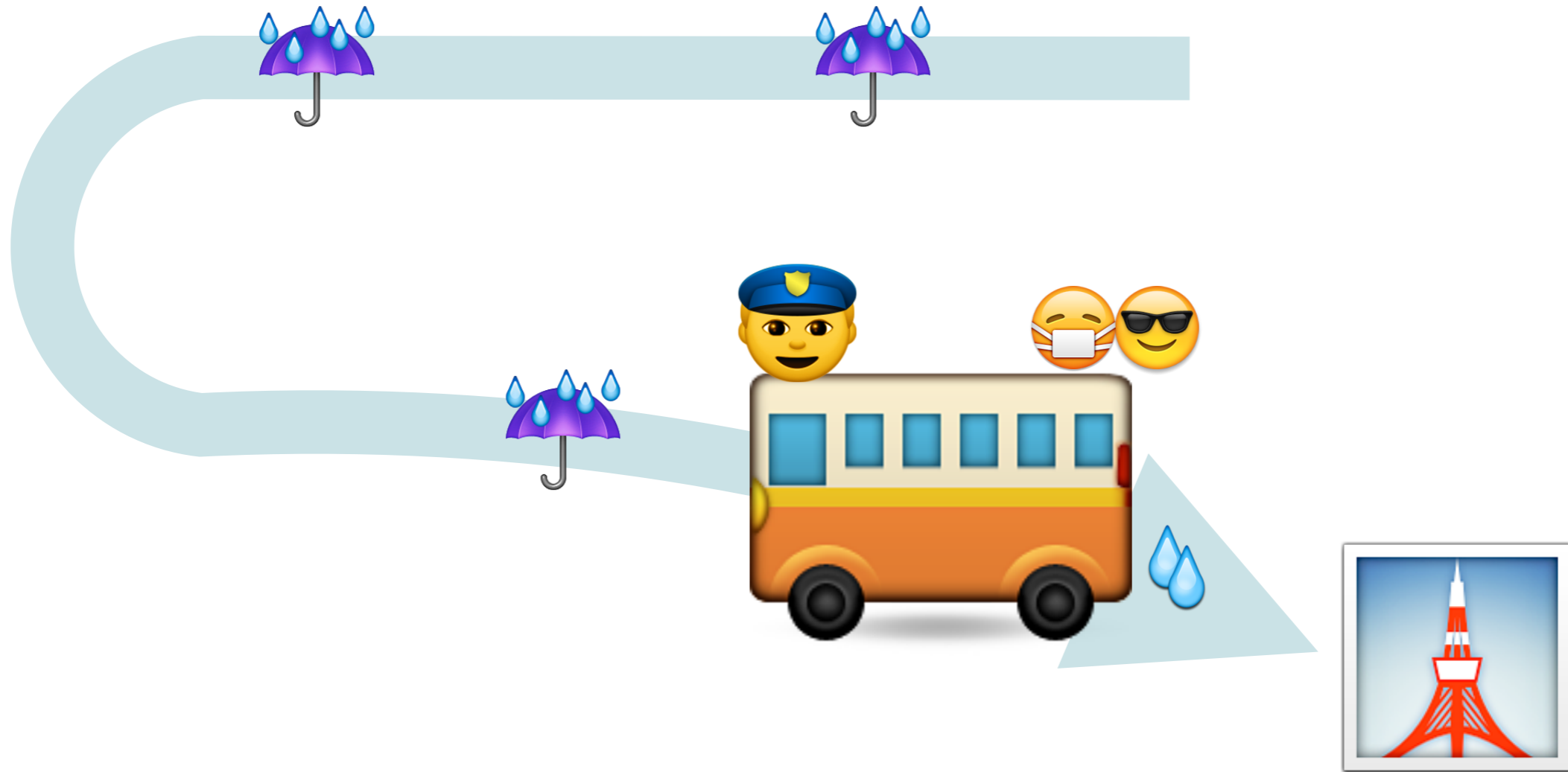
# 問題概要

長距離バスが目的地に行きたい



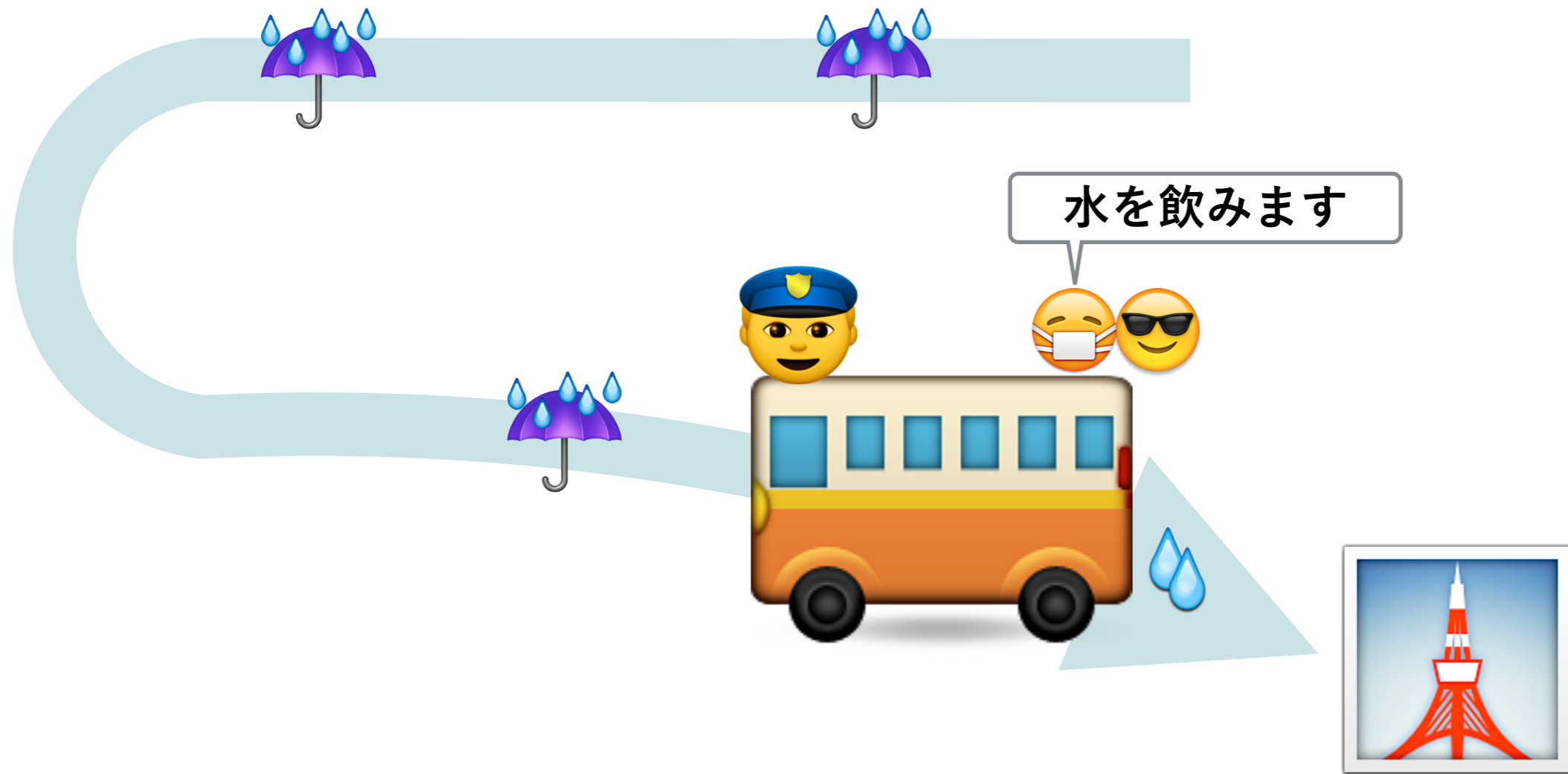
# 問題概要

長距離バスが目的地に行きたい



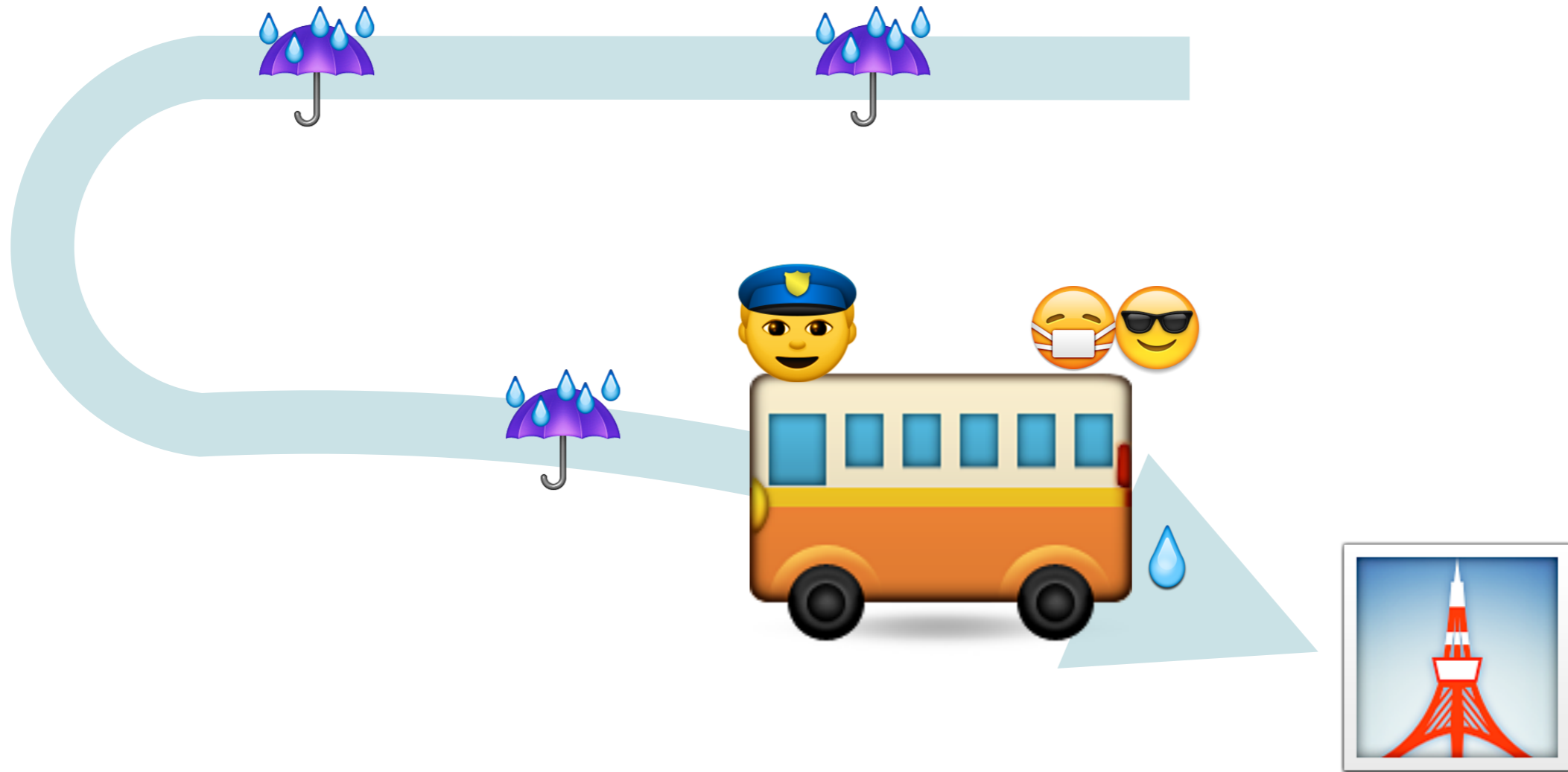
# 問題概要

長距離バスが目的地に行きたい



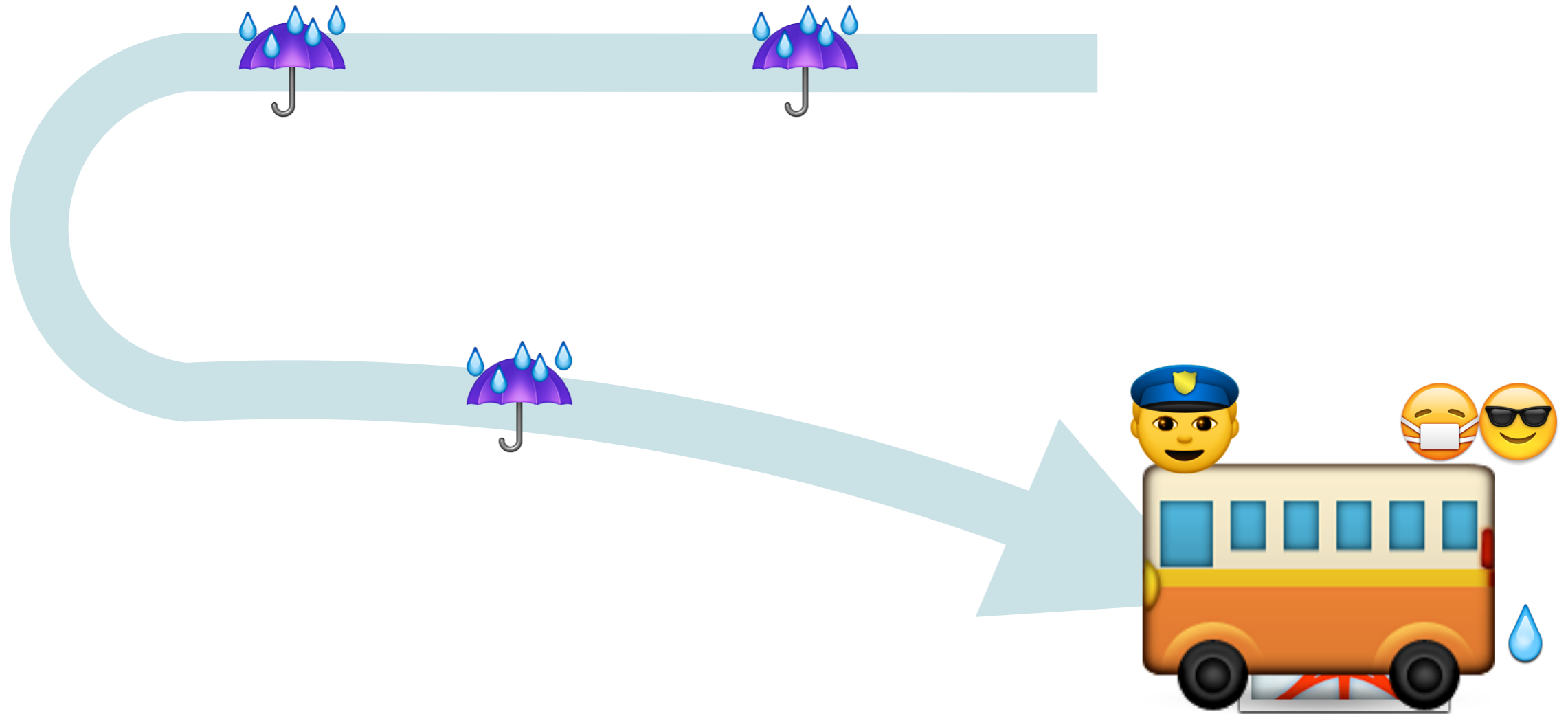
# 問題概要

長距離バスが目的地に行きたい



# 問題概要

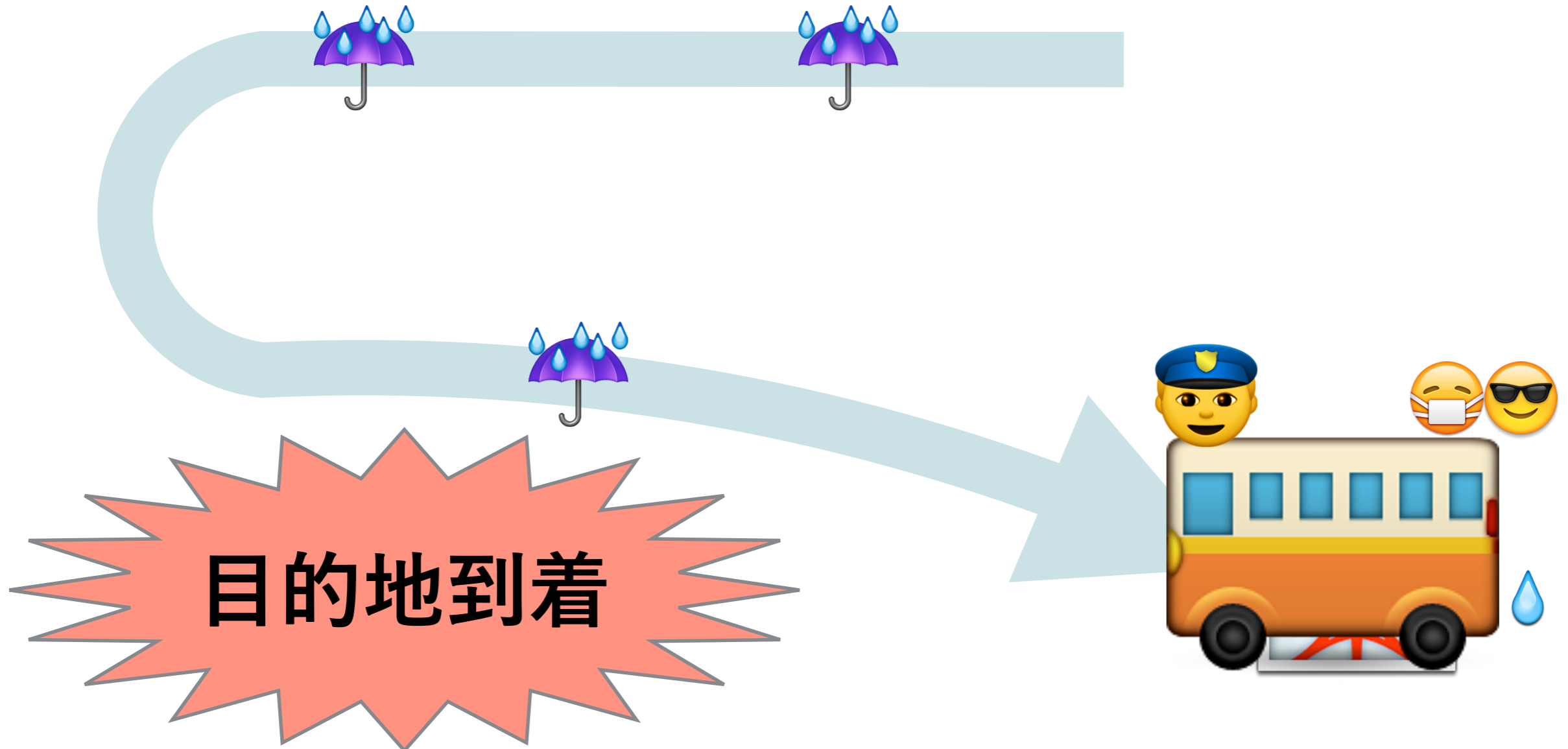
長距離バスが目的地に行きたい





# 問題概要

長距離バスが目的地に行きたい



# 問題概要

---

長距離バスが目的地に行きたい

- ▶ 途中の補給地点で給水器🌧️に💧を補給できる
  - 💧 1個で  $W$  円かかる
- ▶ 運転手👮と乗客😊😌😷😎が**周期的に**💧を飲む
  - 乗客  $j$  は時刻  $D_j + kT$  に💧を 1 個飲む
- ▶ 💧が無いと客は怒って帰る
  - 乗客  $j$  が帰る 😊 → 😡 と  $C_j$  円かかる
  - **運転手が帰ると** 💣 なので回避せねばならない

# 制約

💧 を飲む周期      目的地🗼までの時間

▶  $1 \leq T \leq X \leq 1,000,000,000,000$

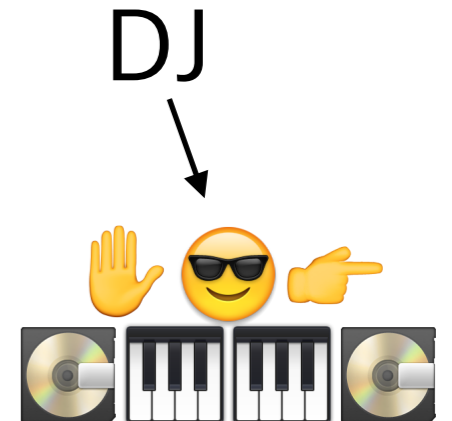
▶  $N \leq 200,000$  ←  ( $N$ 個)

▶  $M \leq 200,000$  ←  ( $M$ 人)

▶  $1 \leq D_j < T$  ←  を最初に飲む時刻

▶  $1 \leq C_j \leq 1,000,000,000$


 →  のコスト



# 小課題 1 (16点)

---

▶  $N \cong 8$

 < はい全探索で常勝で～す

▶  $M \cong 8$

? ? ? ? ? ? ? ? ? ?

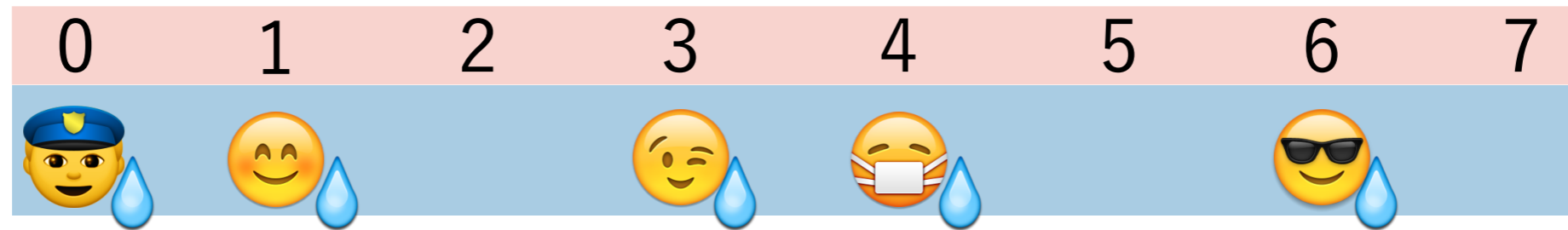
▶  $1 \cong T \cong X \cong 1,000,000,000,000$



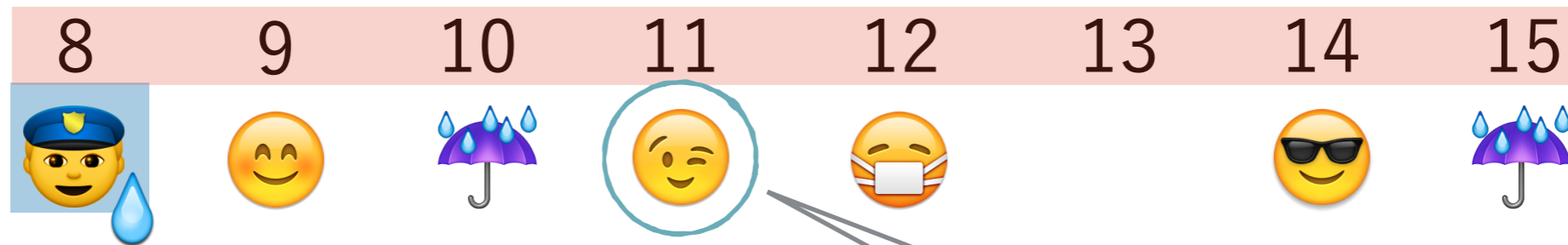
💧 を飲む回数が**数千億回**とか! ??! ?

# 図にする

ここで何個💧を補給するか？ → **6個は必要**



$T$

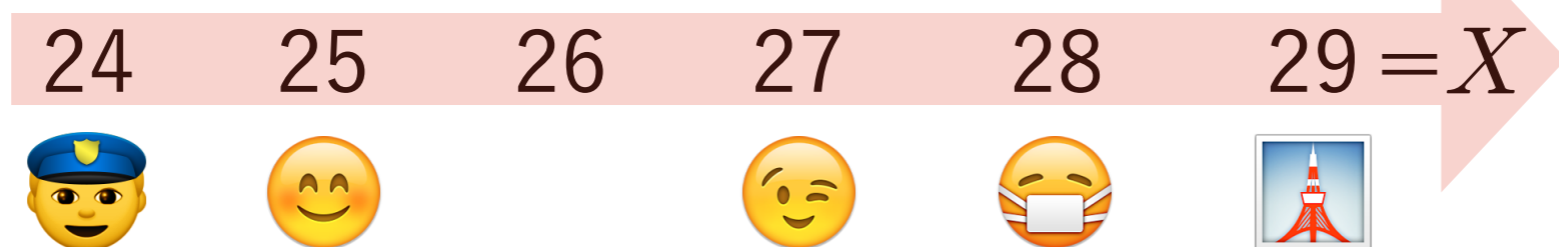


$2T$



この💧を最初に補給する必要はない

$3T$



↓  
**8個は不要**

# 探索

ここで6個補給することにする



	0	1	2	3	4	5	6	7
$T$	8	9	10	11	12	13	14	15
$2T$	16	17	18	19	20	21	22	23
$3T$	24	25	26	27	28	29 = X		

# 探索

ここで6個補給することにする



	0	1	2	3	4	5	6	7
$T$	8	9	10	11	12	13	14	15
$2T$	16	17	18	19	20	21	22	23
$3T$	24	25	26	27	28	29 = X		

# 探索

ここで6個補給することにする



	0	1	2	3	4	5	6	7
$T$	8	9	10	11	12	13	14	15
$2T$	16	17	18	19	20	21	22	23
$3T$	24	25	26	27	28	29 = X		



# 探索

ここで6個補給することにする



# 探索

ここで6個補給することにする



# 探索

---

- ▶ 各🌧️につき補給する💧の個数の候補は  $O(M)$  通り
- ▶ 探索候補は全部で  $O((M+1)^{N+1})$  通り
- ▶ 実際はもっと少ない

😓 2秒で間に合うかなあ……？

実装によっては間に合わないよ 😜



# もうすこしよく見る

この補給地点はこの人を帰らせる権利を持っている



# もうすこしよく見る

この補給地点は



# もうすこしよく見る

この補給地点はこう



# もうすこしよく見る

この補給地点はどうかこう



# もうすこしよく見る

この補給地点はどうかこうかこう





# もうすこしよく見る

この補給地点はこうかこうかこうかこう



$T$

$2T$

$3T$

0 1 2 3 4 5 6 7



8 9 10 11 12 13 14 15



16 17 18 19 20 21 22 23



24 25 26 27 28 29 = X



# 安心できる小課題 1 の解法

---

- ▶ 先にそれぞれの客を帰すか送り届けるか決める
  - ▶ これは  $O(2^M)$  通り
- ▶ すると各🌂で何個💧を補給するか決まる

😓 2秒で間に合うかなあ……？

余裕があるよ 😊



# 問題の言い換え

補給地点 → 帰せる人の区間



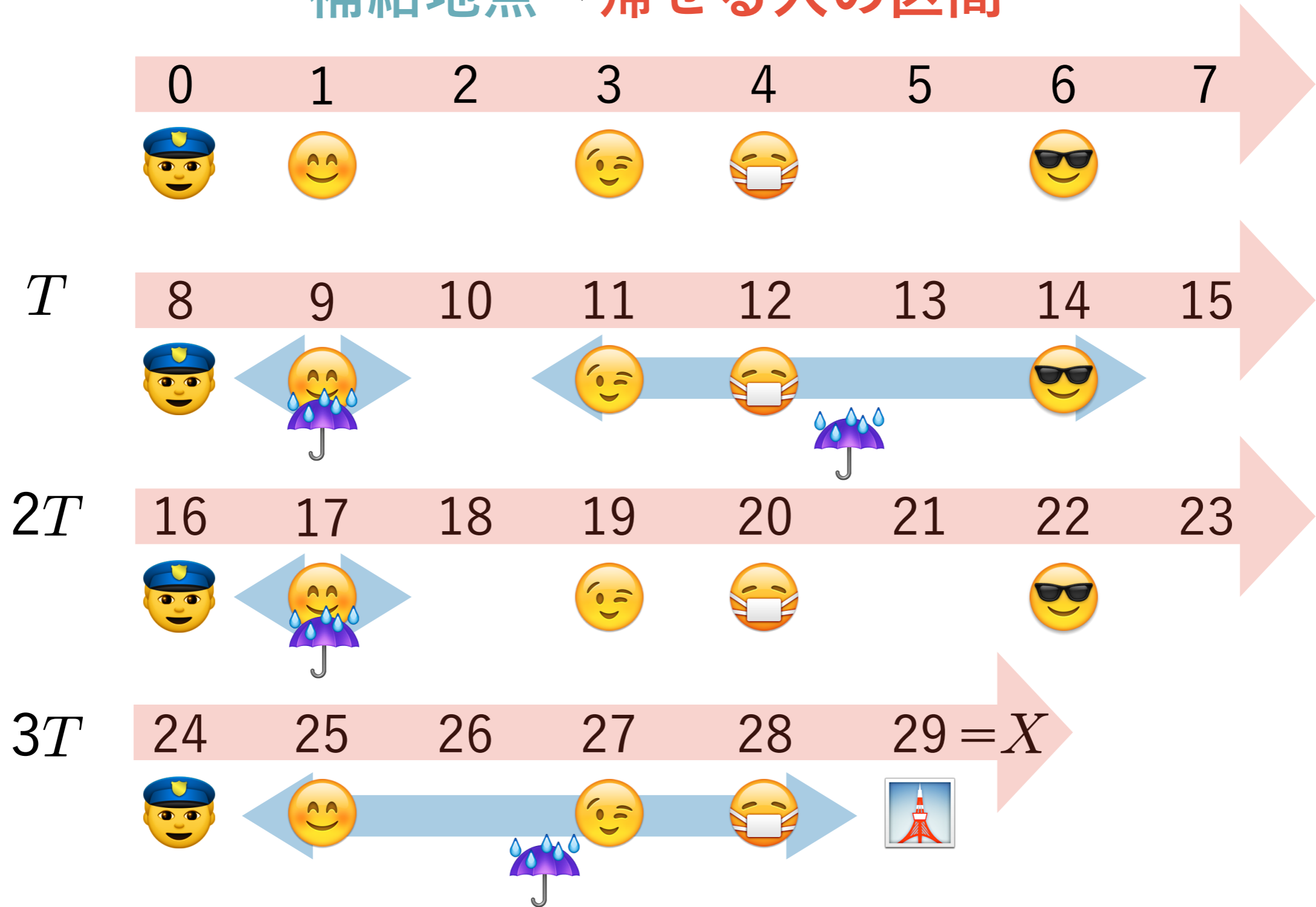
# 問題の言い換え

補給地点 → 帰せる人の区間



# 問題の言い換え

補給地点 → 帰せる人の区間



# 多項式になったで賞 (46点)

---


各🌂を「後ろから連続する何人かを帰せる区間」と

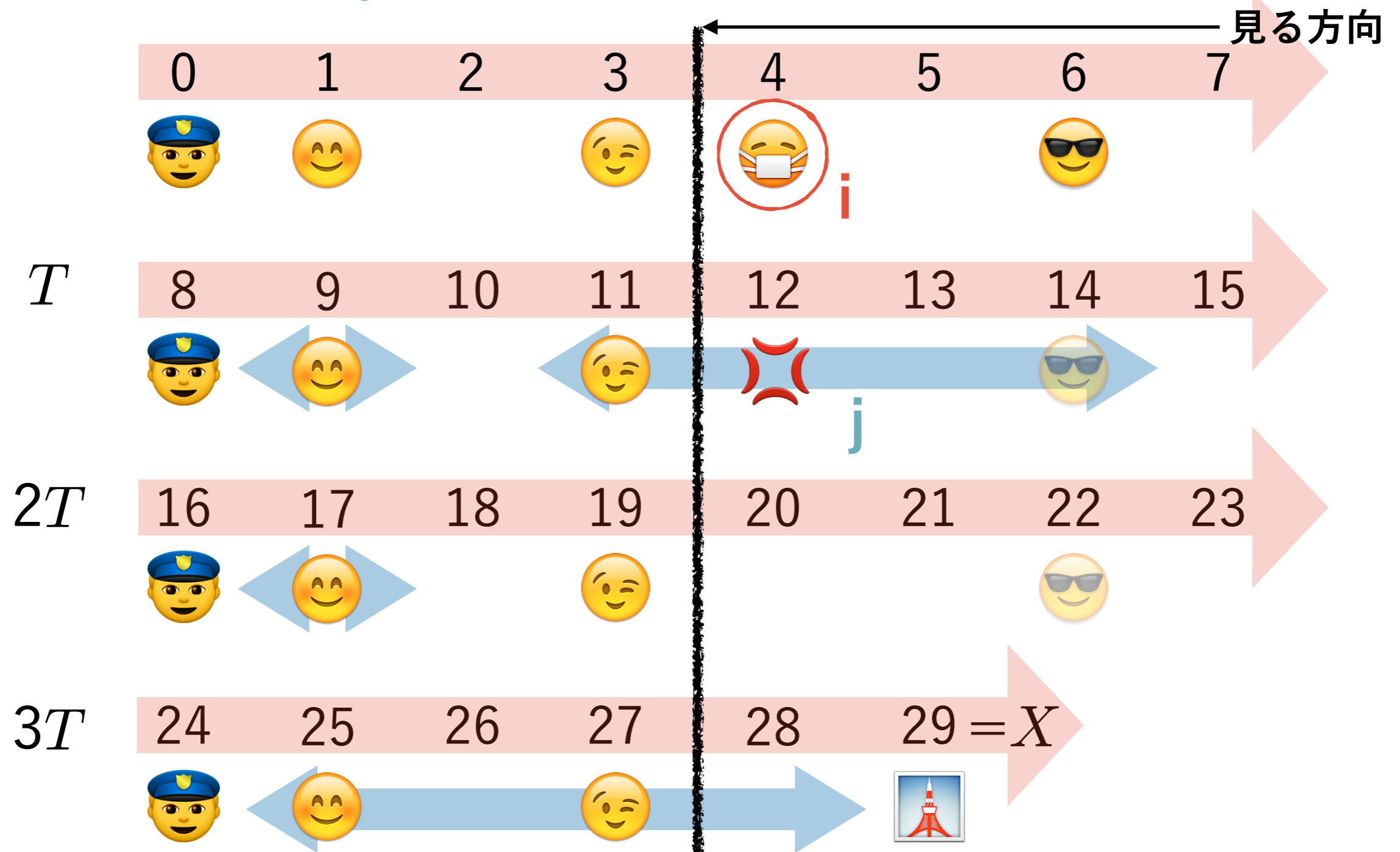
見てなんか DP とかしてください

3 乗以内になれば大丈夫です多分

# ちゃんと DP できたで賞 (71点)


$dp[i][j] :=$  人  $i$  まで見ていて

人  $i$  を   $j$  で追い返したときの最小コスト





# すこし詳しく

$dp[i][j] :=$  人  $i$  まで見ていて

人  $i$  を   $j$  で追い返したときの最小コスト

▶  $dp[i][\text{FREE}] :=$  人  $i$  まで見ていて人  $i$  を送り届ける場合

## 更新式

▶  $dp[i][j] += dp[i + 1][j] +$  (人  $i$  を   $j$  で帰すコスト)  
-   $j$  がすでにある場合

▶  $dp[i][j] = \min\{dp[i + 1][\star]\} +$  (人  $i$  を   $j$  で帰す)  
▶   $j$  が人  $i$  と  $i+1$  の間で出てくる場合

▶  $dp[i][\text{FREE}] = \min\{dp[i + 1][\star]\} +$  (人  $i$  を送り届ける)






# すこし詳しく


$dp[i][j] :=$  人  $i$  まで見ていて


人  $i$  を   $j$  で追い返したときの最小コスト

▶  $dp[i][\text{FREE}] :=$  人  $i$  まで見ていて人  $i$  を送り届ける場合

## 更新式

▶  $dp[i][j] += dp[i + 1][j] +$  (人  $i$  を   $j$  で帰すコスト)

-   $j$  がすでにある場合

▶  $dp[i][j] = \min\{dp[i + 1][\star]\} +$  (人  $i$  を   $j$  で帰す)

▶   $j$  が人  $i$  と  $i+1$  の間で出てくる場合

▶  $dp[i][\text{FREE}] = \min\{dp[i + 1][\star]\} +$  (人  $i$  を送り届ける)







# すこし詳しく


$dp[i][j] :=$  人  $i$  まで見ていて

人  $i$  を   $j$  で追い返したときの最小コスト

▶  $dp[i][\text{FREE}] :=$  人  $i$  まで見ていて人  $i$  を送り届ける場合

## 更新式

▶  $dp[i][j] += dp[i + 1][j] +$  (人  $i$  を   $j$  で帰すコスト)  
-   $j$  がすでにある場合


▶  $dp[i][j] = \min\{dp[i + 1][\star]\} +$  (人  $i$  を   $j$  で帰す)


▶   $j$  が人  $i$  と  $i+1$  の間で出てくる場合

▶  $dp[i][\text{FREE}] = \min\{dp[i + 1][\star]\} +$  (人  $i$  を送り届ける)

# DP 早くなつたで賞 (100点)

DP の更新式をよく見てみよう

人 \ 	FREE	1	2	3	4	5	6	7	8
9	*	*	*	*	$\infty$	*	*	*	*
8	*	*	*	*	$\infty$	*	*	*	*
7	*	*	*	*	$\infty$	*	*	*	*
6									
5									
4									
3									
2									
1									

人 6 と 7 の間で  
 4 が始まる

# DP 早くなっただで賞 (100点)

DP の更新式をよく見てみよう


人 \ 🌧️	FREE	1	2	3	4	5	6	7	8
9	*	*	*	*	∞	*	*	*	*
8	*	*	*	*	∞	*	*	*	*
7	*	*	*	*	∞	*	*	*	*
6									
5									
4									
3									
2									
1									

min をとる

人 6 と 7 の間で  
🌧️ 4 が始まる

# すこし詳しく


$dp[i][j] :=$  人  $i$  まで見ていて

人  $i$  を   $j$  で追い返したときの最小コスト

▶  $dp[i][\text{FREE}] :=$  人  $i$  まで見ていて人  $i$  を送り届ける場合

## 更新式

▶  $dp[i][j] += dp[i + 1][j] + (\text{人 } i \text{ を } \img alt="purple umbrella icon" data-bbox="600 535 635 580"/>  $j$  で帰すコスト)$

-   $j$  がすでにある場合

▶  $dp[i][j] = \min\{dp[i + 1][\star]\} + (\text{人 } i \text{ を } \img alt="purple umbrella icon" data-bbox="685 695 720 740"/>  $j$  で帰す)$


▶   $j$  が人  $i$  と  $i+1$  の間で出てくる場合


▶  $dp[i][\text{FREE}] = \min\{dp[i + 1][\star]\} + (\text{人 } i \text{ を送り届ける})$



# DP 早くなっただで賞 (100点)

DP の更新式をよく見てみよう

人 \ 	FREE	1	2	3	4	5	6	7	8
9	*	*	*	*	$\infty$	*	*	*	*
8	*	*	*	*	$\infty$	*	*	*	*
7	*	*	*	*	$\infty$	*	*	*	*
6	*	*	*	*	*	*	*	*	*
5									
4									
3									
2									
1									

人5を  4で帰すコスト

# DP 早くなつたで賞 (100点)

DP の更新式をよく見てみよう

人 \ 傘	FREE	1	2	3	4	5	6	7	8
9	*	*	*	*	$\infty$	*	*	*	*
8	*	*	*	*	$\infty$	*	*	*	*
7	*	*	*	*	$\infty$	*	*	*	*
6	*	*	*	*	*	*	*	*	*
5					*				
4					*				
3									
2									
1									

人5を傘4で帰すコスト

人4を傘4で帰すコスト

# DP 早くなっただで賞 (100点)

DP の更新式をよく見てみよう


人 \ 傘	FREE	1	2	3	4	5	6	7	8
9	*	*	*	*	$\infty$	*	*	*	*
8	*	*	*	*	$\infty$	*	*	*	*
7	*	*	*	*	$\infty$	*	*	*	*
6	*	*	*	*	*	*	*	*	*
5					*				
4					*				
3									
2									
1									


人5を傘4で帰すコスト


人4を傘4で帰すコスト


他の列によらず決まる

# DP 早くなつたで賞 (100点)

人  $i$  を   $j$  で帰すコスト =  $W * (S[j]/T) + C[i]$   
人 に 依 存 し な い      人 に 依 存


人 \ 	FREE	1	2	3	4	5	6	7	8
9	*	*	*	*	$\infty$	*	*	*	*
8	*	*	*	*	$\infty$	*	*	*	*
7	*	*	*	*	$\infty$	*	*	*	*
6	*	*	*	*	*	*	*	*	*
5					*				
4					*				
3									
2									
1									

人5を  4 で帰すコスト

人4を  4 で帰すコスト

他の列によらず決まる


# DP 早くなっただで賞 (100点)


人  $i$  を   $j$  で帰すコスト =  $W * (S[j]/T) + C[i]$   
人 に 依 存 し ない      人 に 依 存

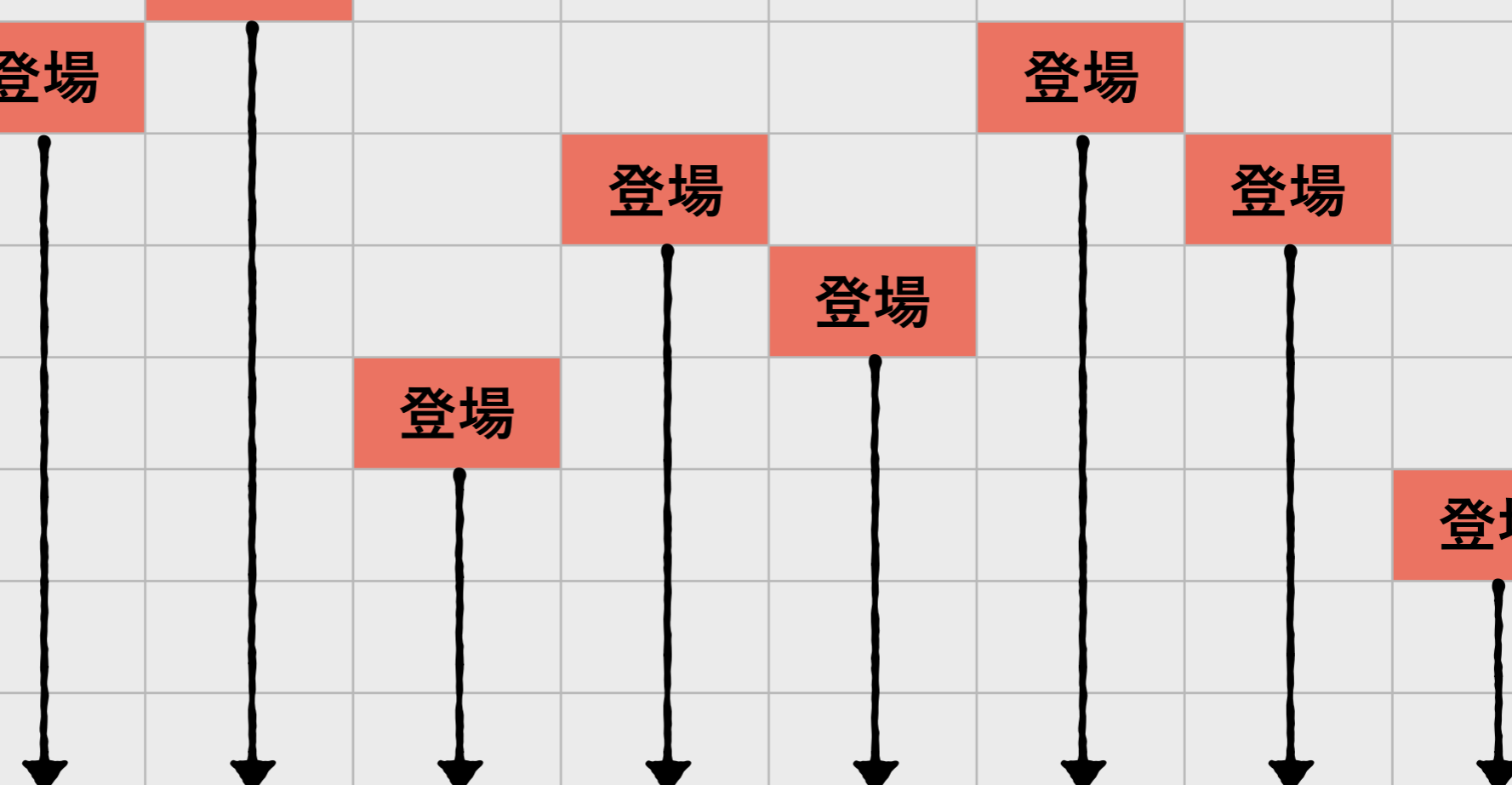
人 \ 	FREE	1	2	3	4	5	6	7	8
9									
8									
7	あらかじめ足されていると思っておく (min に影響はないことに注意)								
6									
5									
4					⋮				
3				+C[3]+⋯+C[9]					
2				+C[2]+⋯+C[9]					
1				+C[1]+⋯+C[9]					



# DP 早くなつたで賞 (100点)

人  $i$  を   $j$  で帰すコスト =  $W * (S[j]/T)$  +  $C[i]$   
人 に 依 存 し な い      人 に 依 存


人 \ 	FREE	1	2	3	4	5	6	7	8
9									
8			登場						
7		登場					登場		
6					登場			登場	
5						登場			
4				登場					
3									登場
2									
1									








# DP 早くなっただで賞 (100点)

人  $i$  を   $j$  で帰すコスト =  $W * (S[j]/T) + C[i]$   
人 に 依 存 し ない      人 に 依 存

人 \ 	FREE	1	2	3	4	5	6	7	8
9									
8			登場						
7		登場					登場		
6					登場			登場	
5						登場			
4				登場					
3									登場
2									
1									

直線がたくさん!

min がとりたい

# Convex Hull Trick


# Convex Hull Trick

---

▶ 汎用テク & バリエーションも多いので詳細は各自

- 蟻本にもある？ ネット上にも結構解説がある

▶ 要らない直線を管理しつつ頑張っていけば……

 いやあ大変だったなあ……

? ? ? ? ? ? ? ? ? ?

▶  $1 \leq T \leq X \leq 1,000,000,000,000$



# 要らない直線の判定法

---

▶ 3 直線  $y_i = a_i x + b_i$  ( $i=1,2,3$ ) が  $a_1 > a_2 > a_3$  のとき

$$(a_2 - a_1)(b_3 - b_2) \cong (b_2 - b_1)(a_3 - a_2)$$

? ? ? ? ? ? ? ? ? ?

▶  $1 \cong T \cong X \cong 1,000,000,000,000$



# 世はまさに大オーバーフロー時代

---

- ▶ 回避方法1: 頑張る

- ▶  $a/d \geq b/c$  を判定しようとする

- ▶ 整数部分が違うなら簡単で

- 同じ時は mod をとって小数部分を再帰比較する

- ▶ 回避方法2: doubleる

- ▶ どうせそんなきわどいケースないでしょ……

- ▶ 回避方法3: `__int128`

- ▶ パソコンに詳しい人向け

# 得点分布

