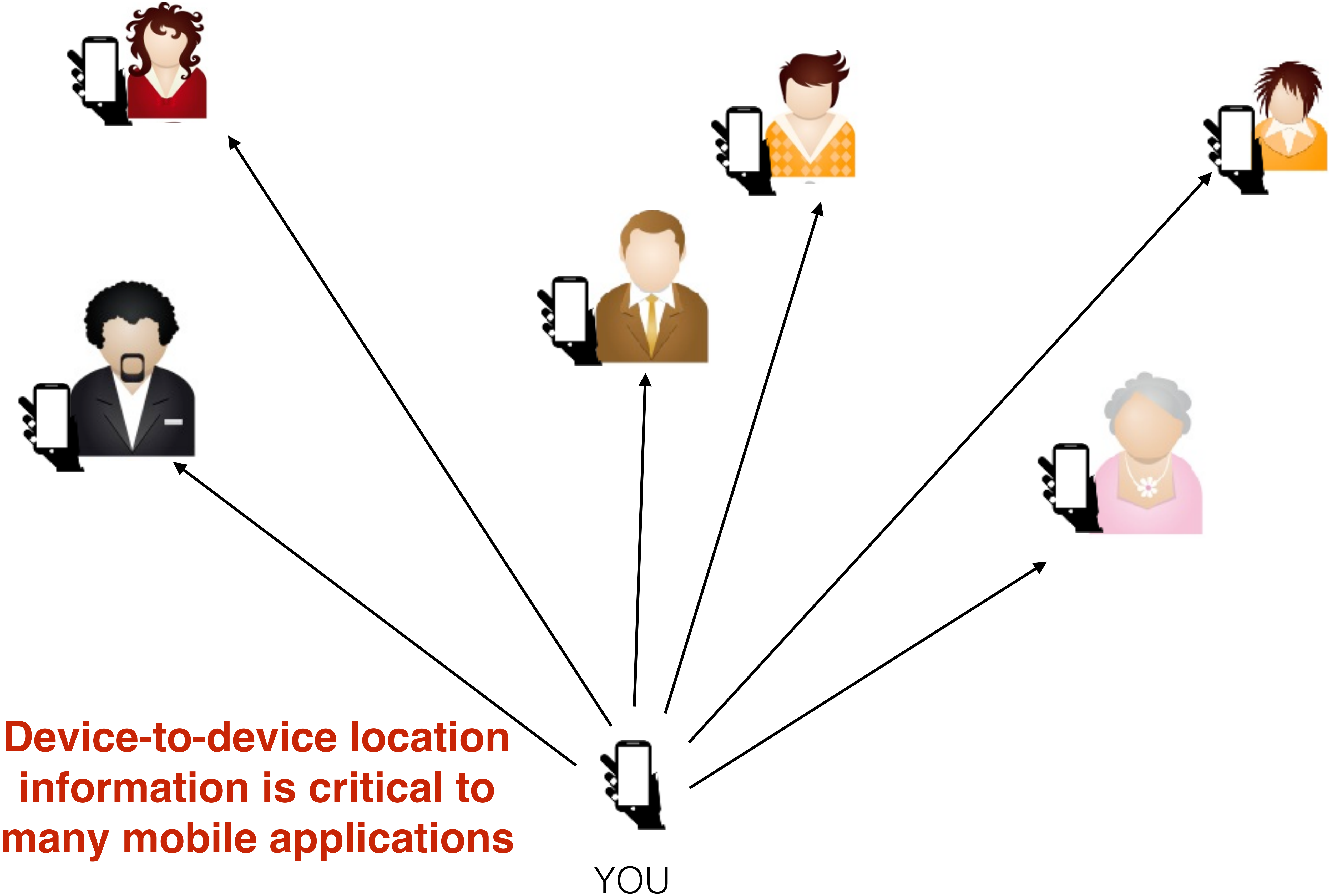
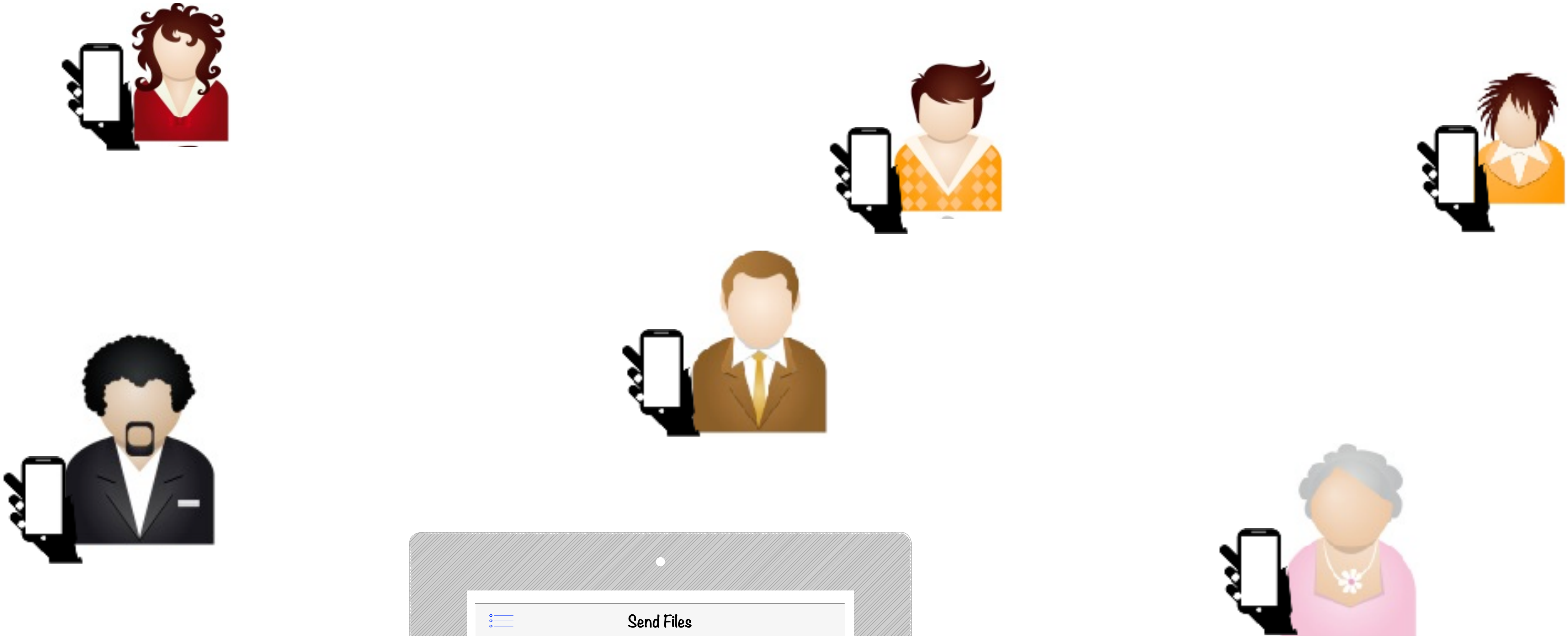
AMIL: Localizing Neighboring Mobile Devices Through a Simple Gesture

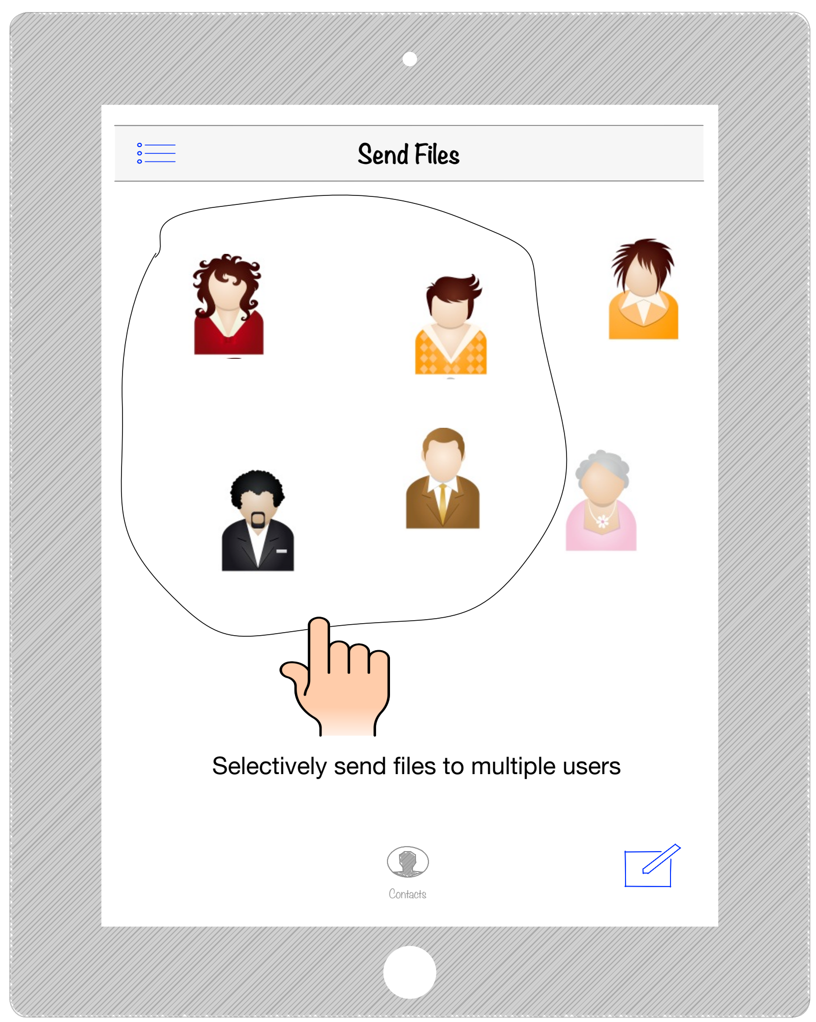
Hao Han, Shanhe Yi*, Qun Li*, Guobin Shen**, Yunxin Liu** and Ed Novak*

Intelligent Automation, Inc.
*College of William and Mary
**Microsoft Research Asia



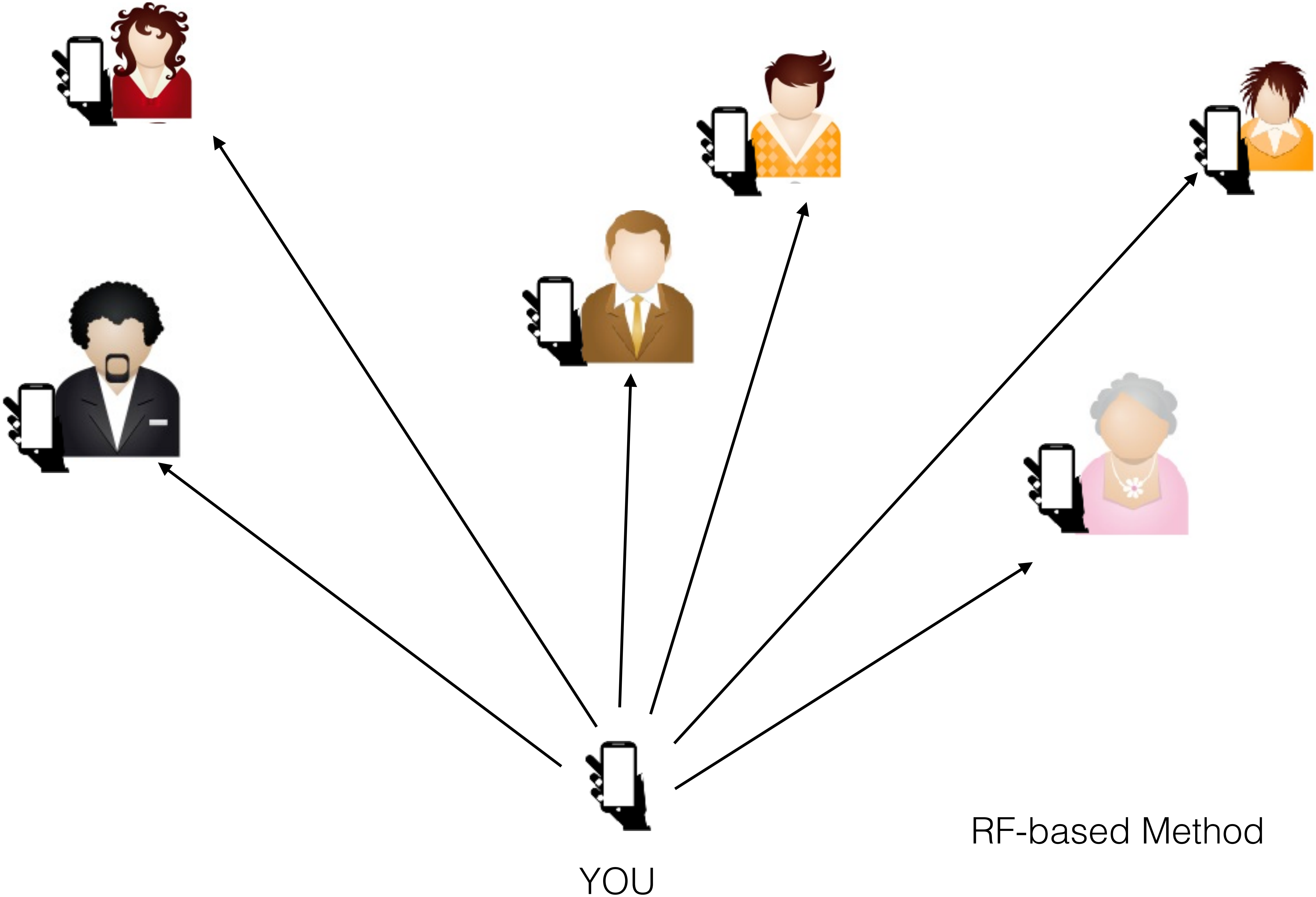
Device-to-device location information is critical to many mobile applications





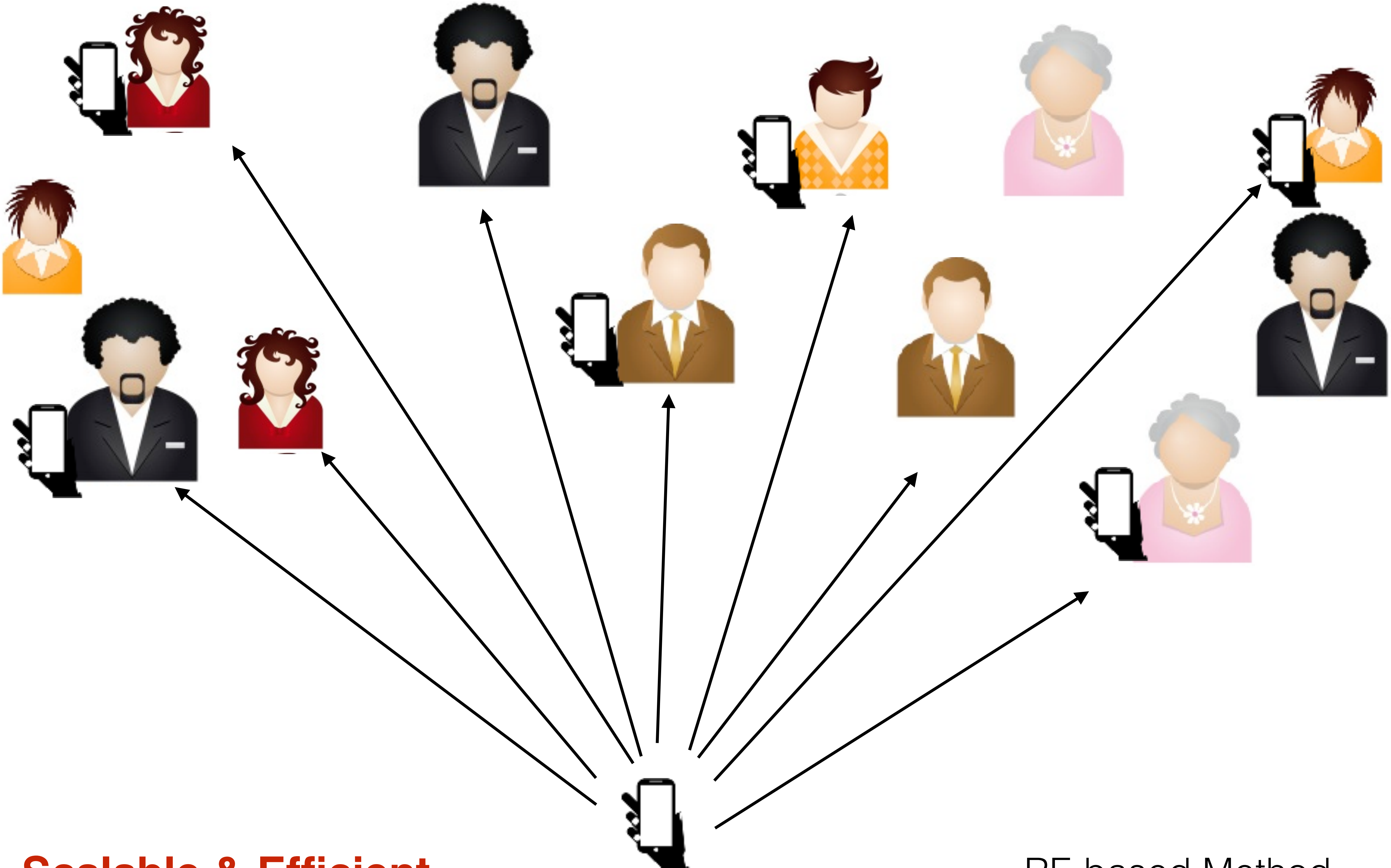
Location Info can help.

- Exchange contacts
- Send documents
- Create groups
- Multiplayer games
- and many more …



RF-based Method

Acoustic-based Method

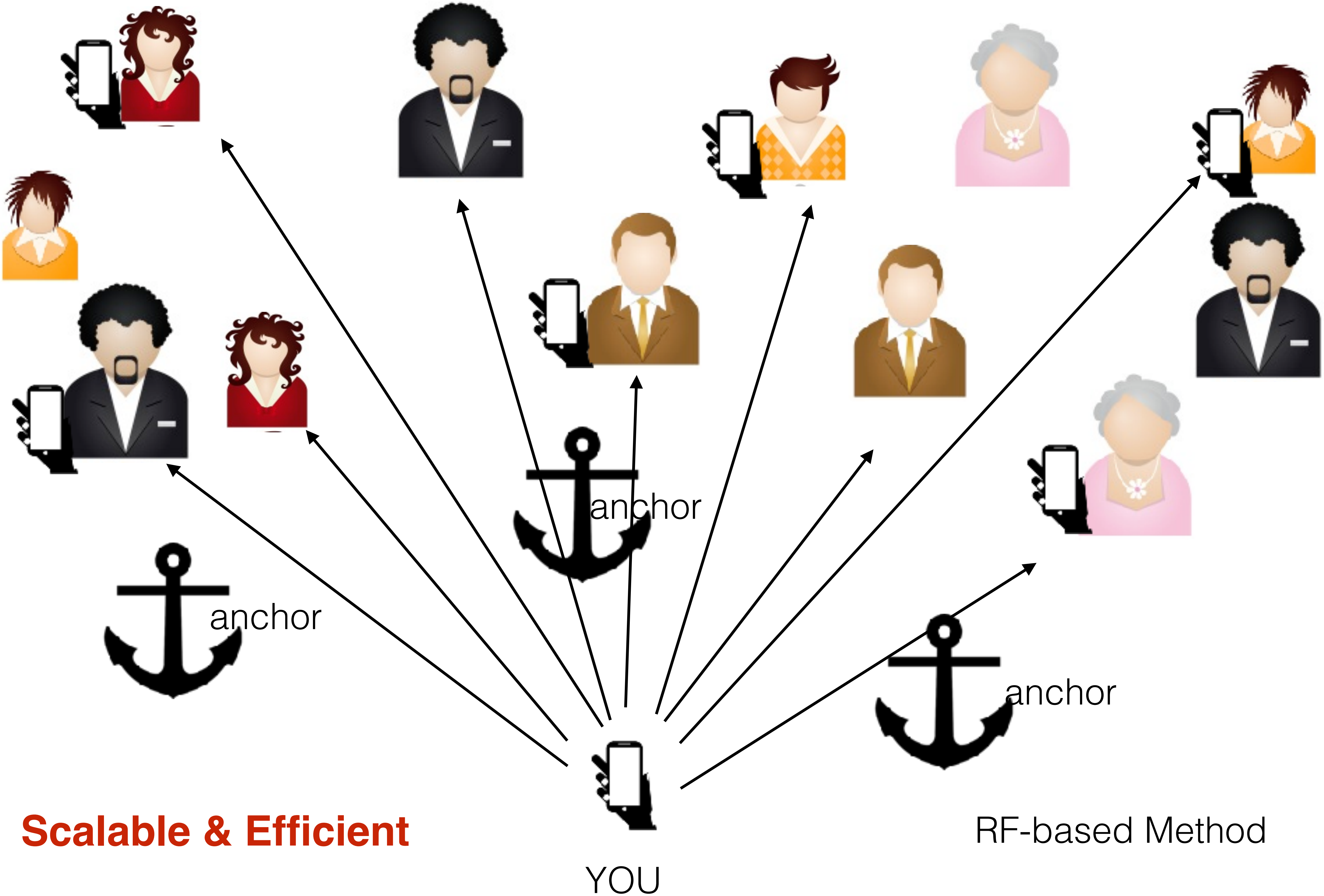


YOU

Scalable & Efficient

RF-based Method

Acoustic-based Method

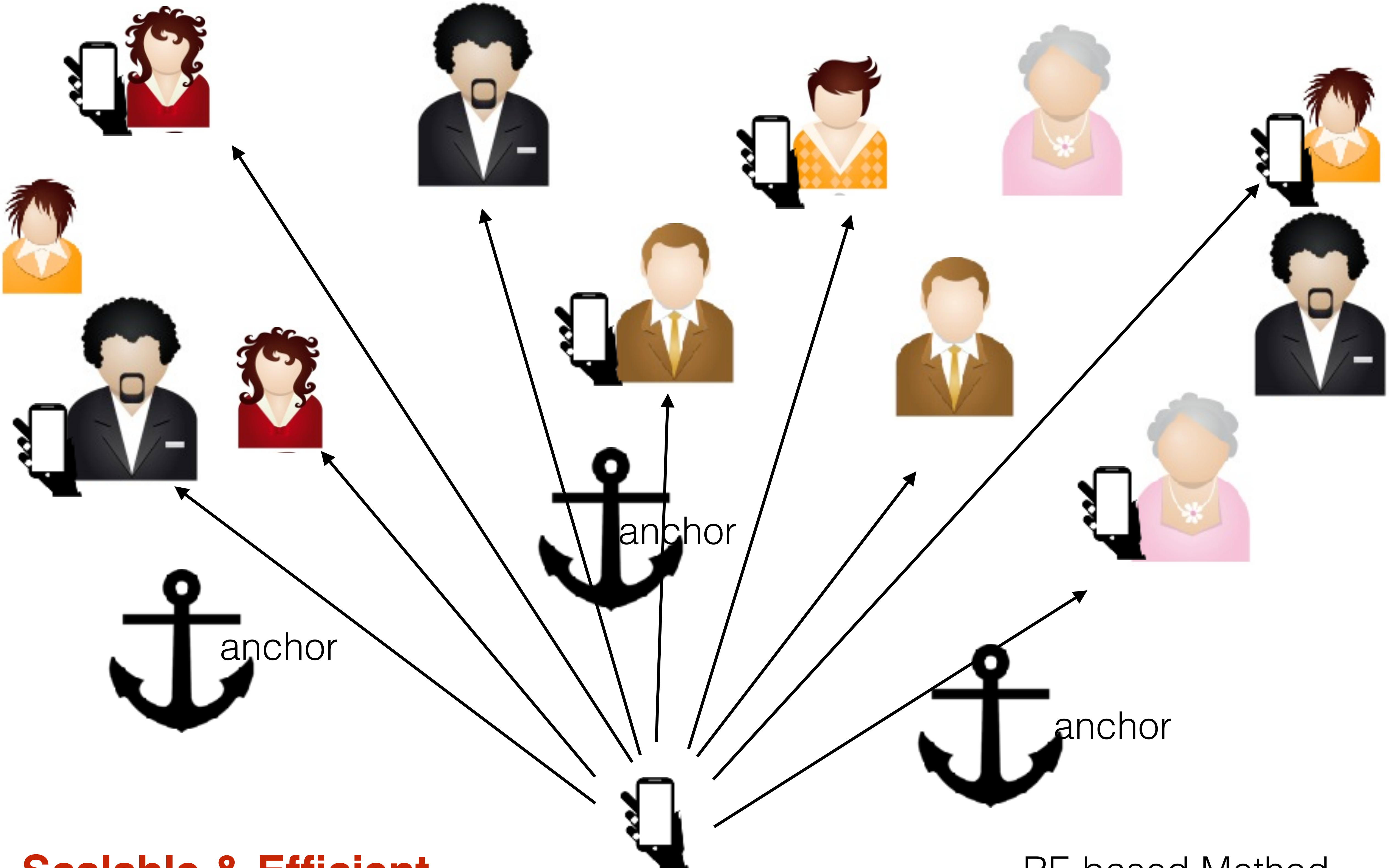


Scalable & Efficient

Light weight, Infrastructure-free

RF-based Method

Acoustic-based Method



YOU

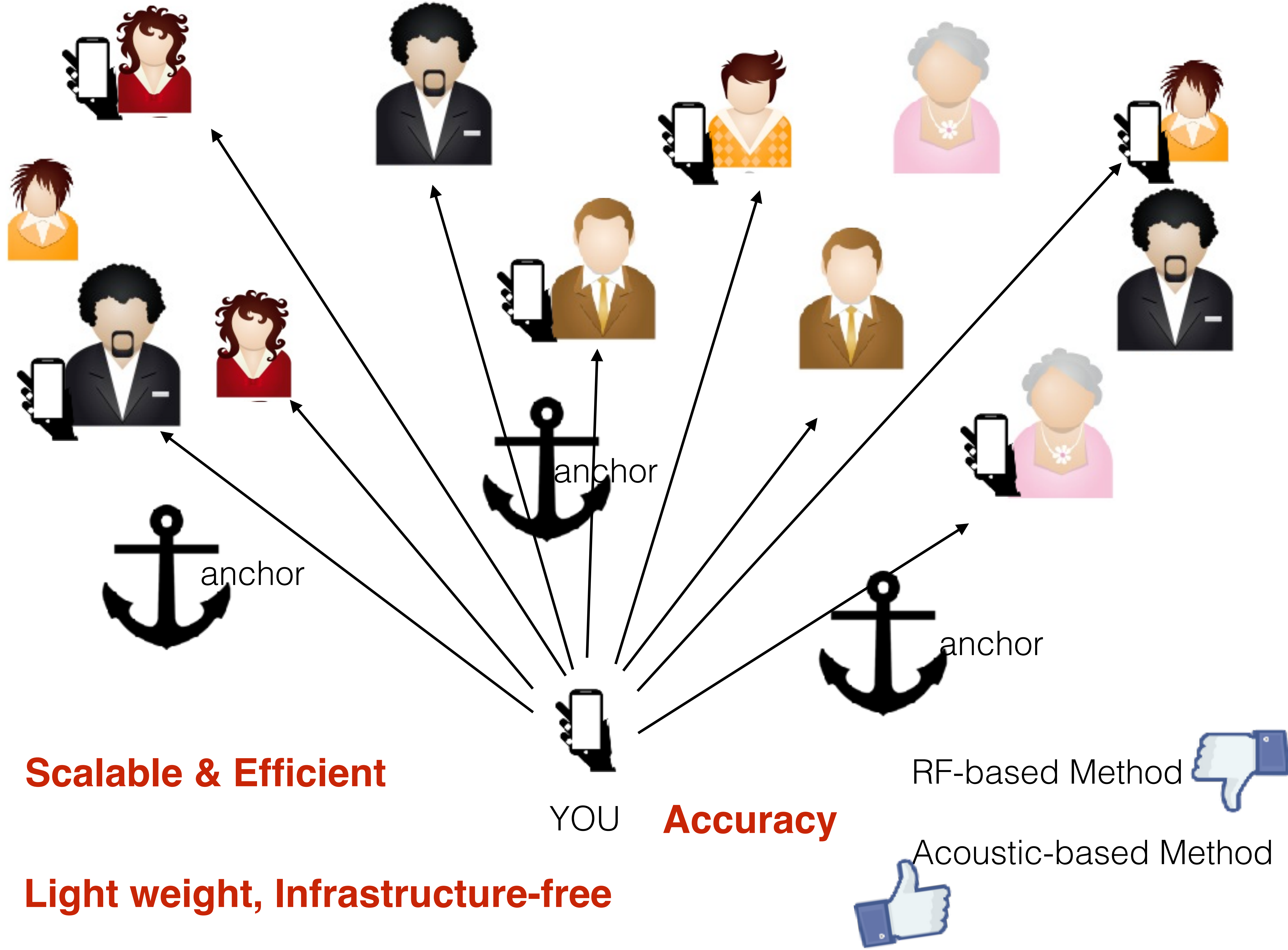
Scalable & Efficient

Accuracy

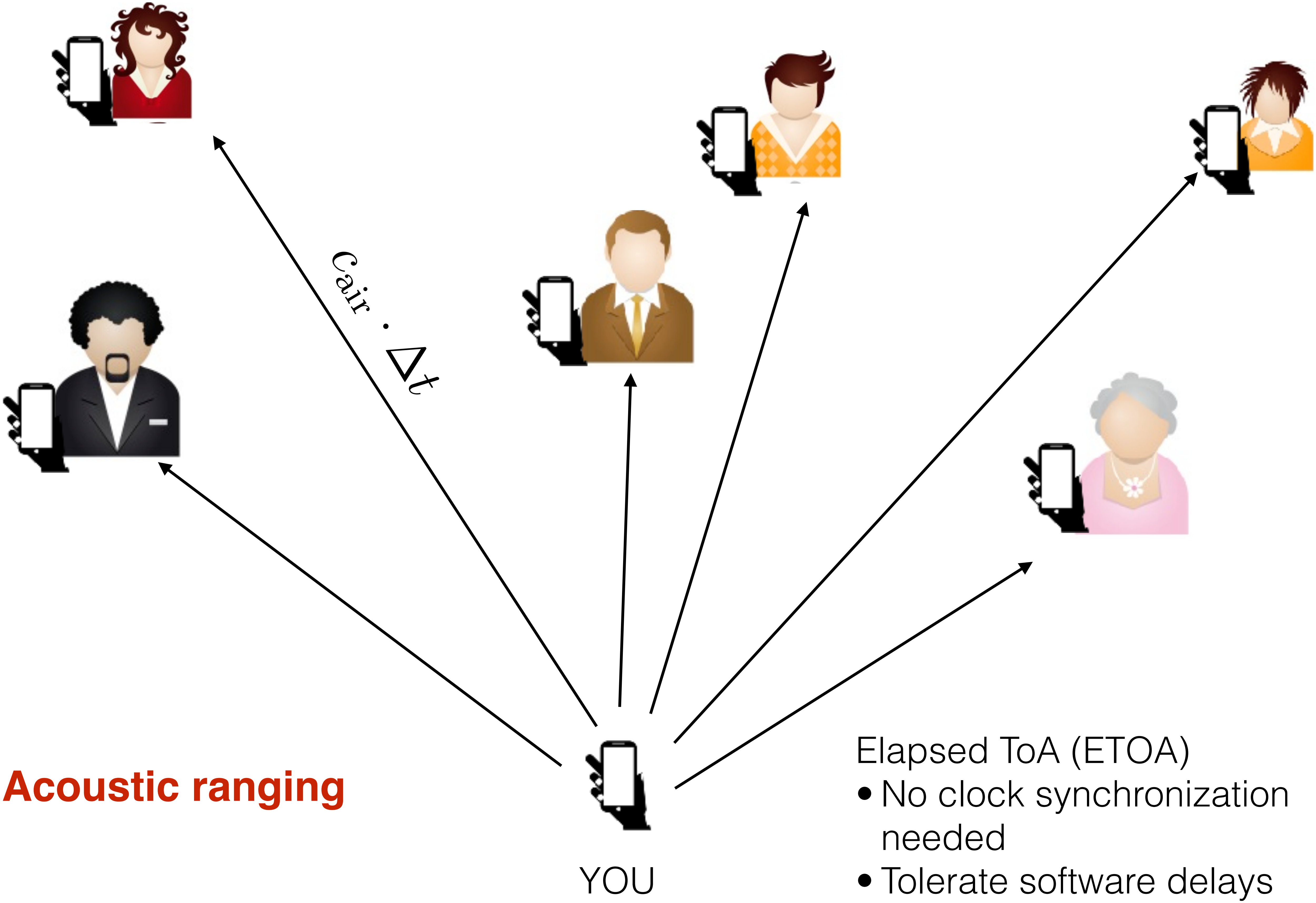
RF-based Method

Acoustic-based Method

Light weight, Infrastructure-free


anchor
anchor
anchor
YOU
Scalable & Efficient
Accuracy
Light weight, Infrastructure-free
RF-based Method
Acoustic-based Method

Acoustic ranging

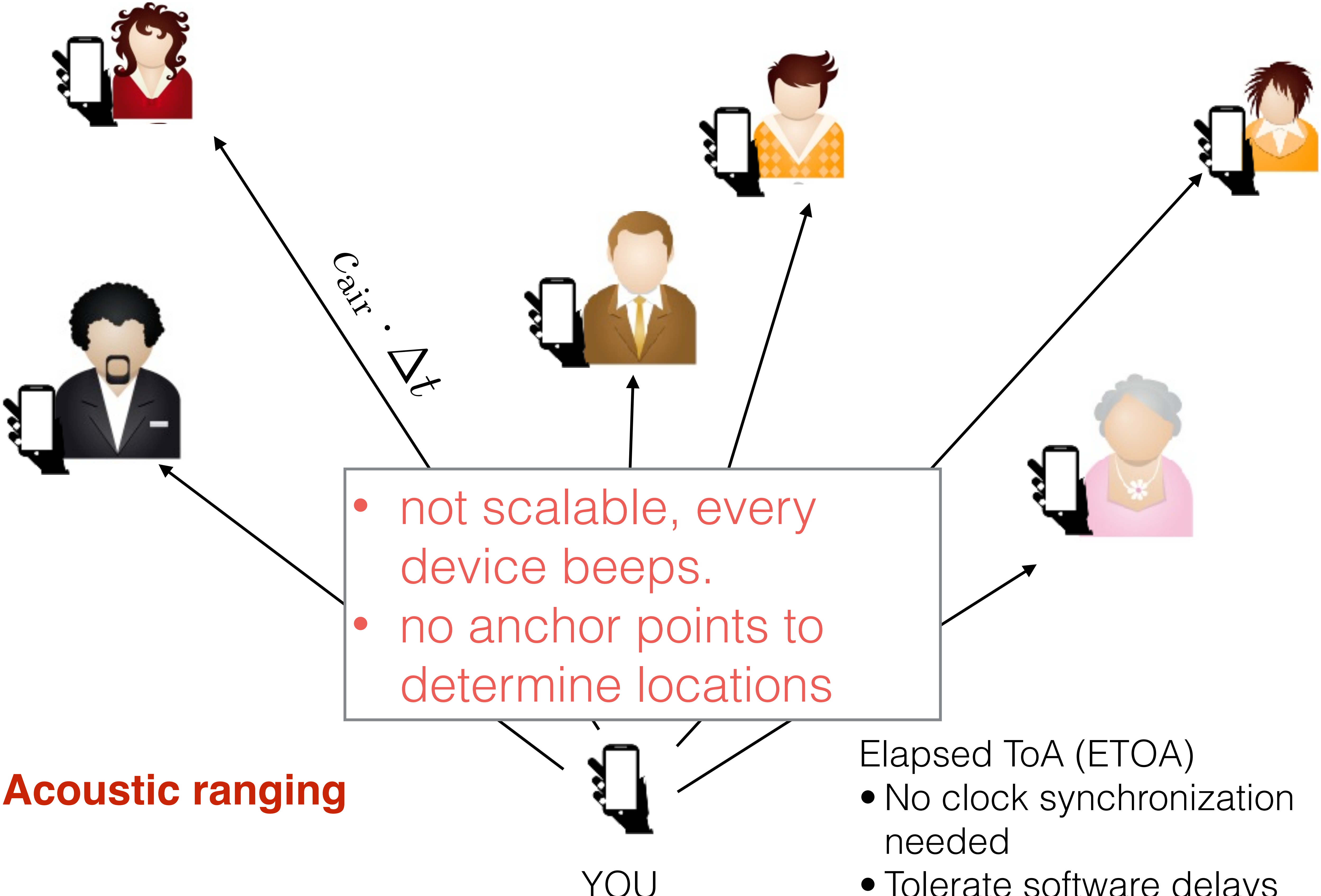


Elapsed ToA (ETOA)

- No clock synchronization needed
- Tolerate software delays

Peng *et. al*. BeepBeep: A High Accuracy Acoustic Ranging System using COTS Mobile Devices, Sensys'07

Acoustic ranging

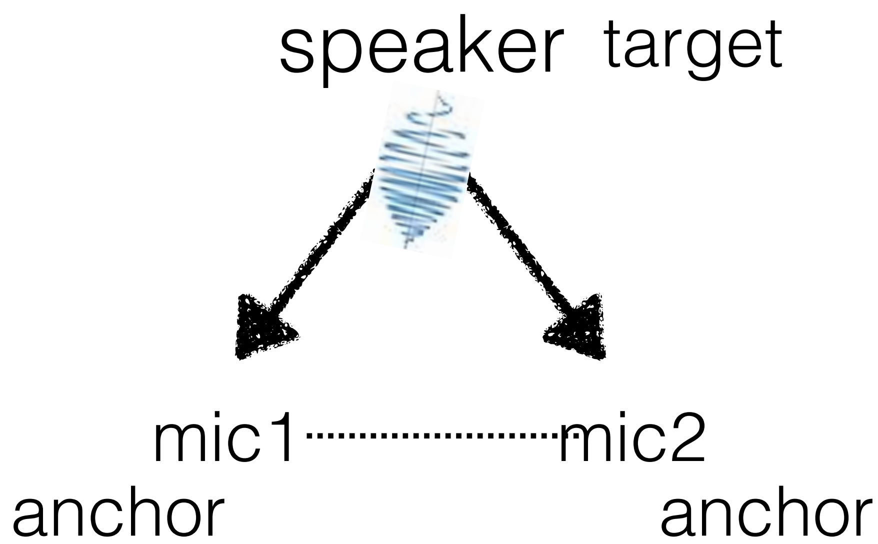


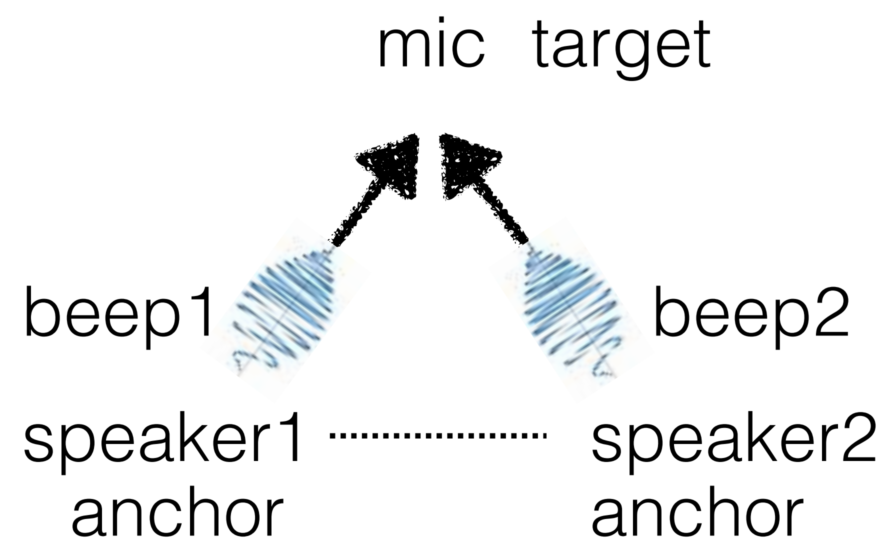
- not scalable, every device beeps.
- no anchor points to determine locations

Elapsed ToA (ETOA)
- No clock synchronization needed
- Tolerate software delays

Peng *et. al*. BeepBeep: A High Accuracy Acoustic Ranging System using COTS Mobile Devices, Sensys'07

A prime on Time-Difference-of-Arrival

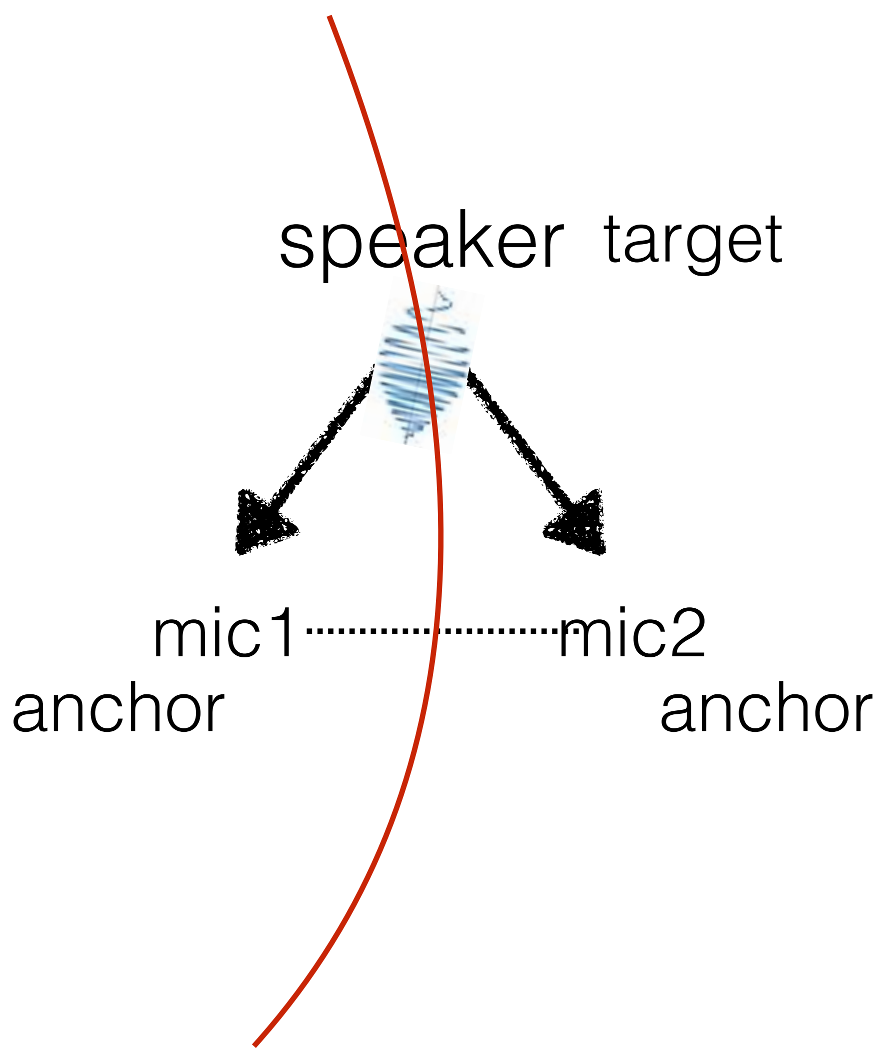




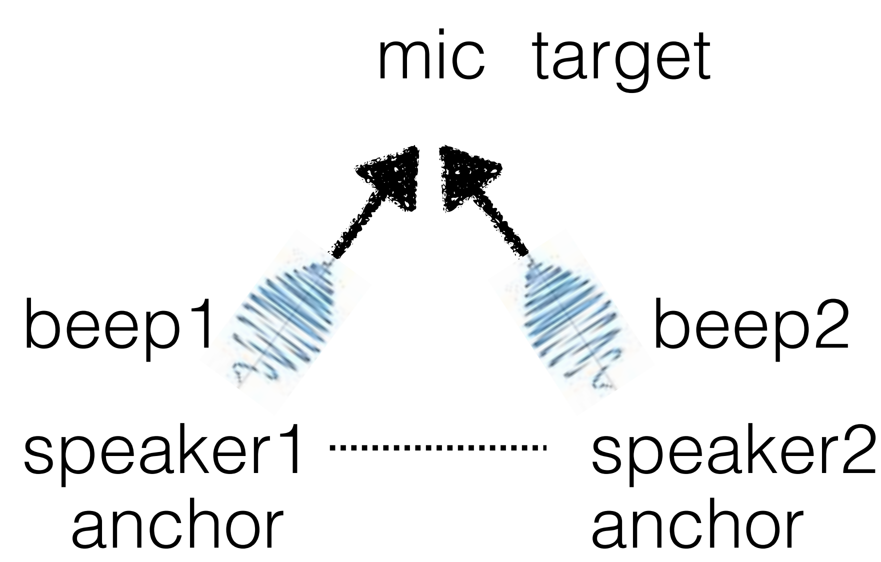
Classic TDoA

A variant of TDoA

A prime on Time-Difference-of-Arrival

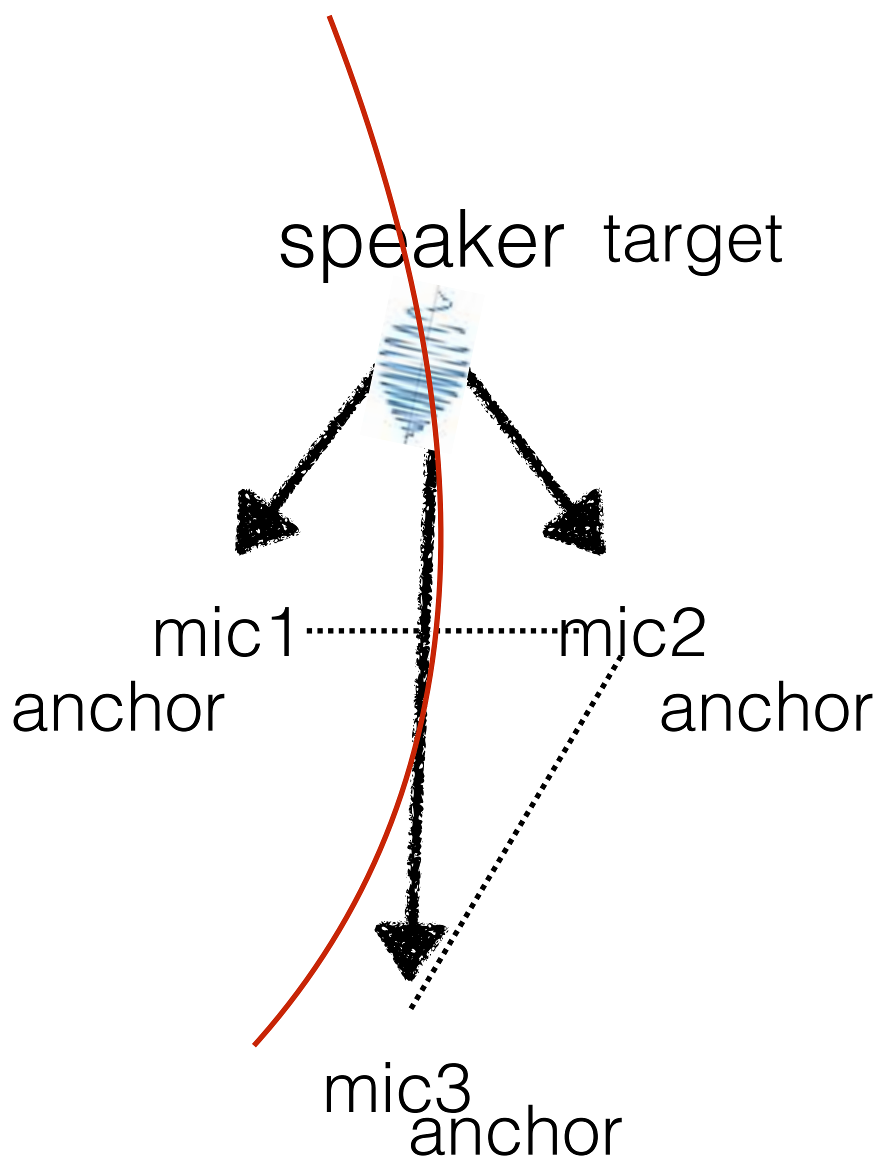


Classic TDoA

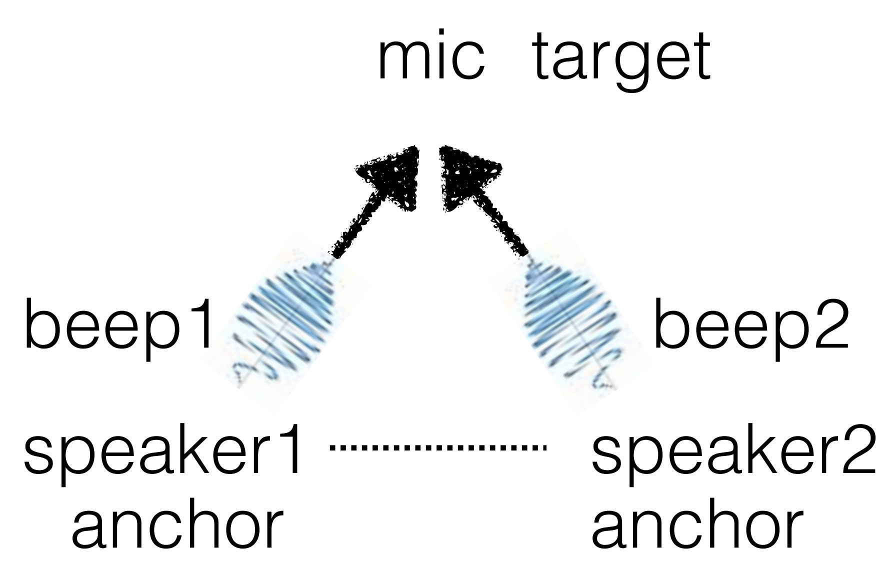


A variant of TDoA

A prime on Time-Difference-of-Arrival

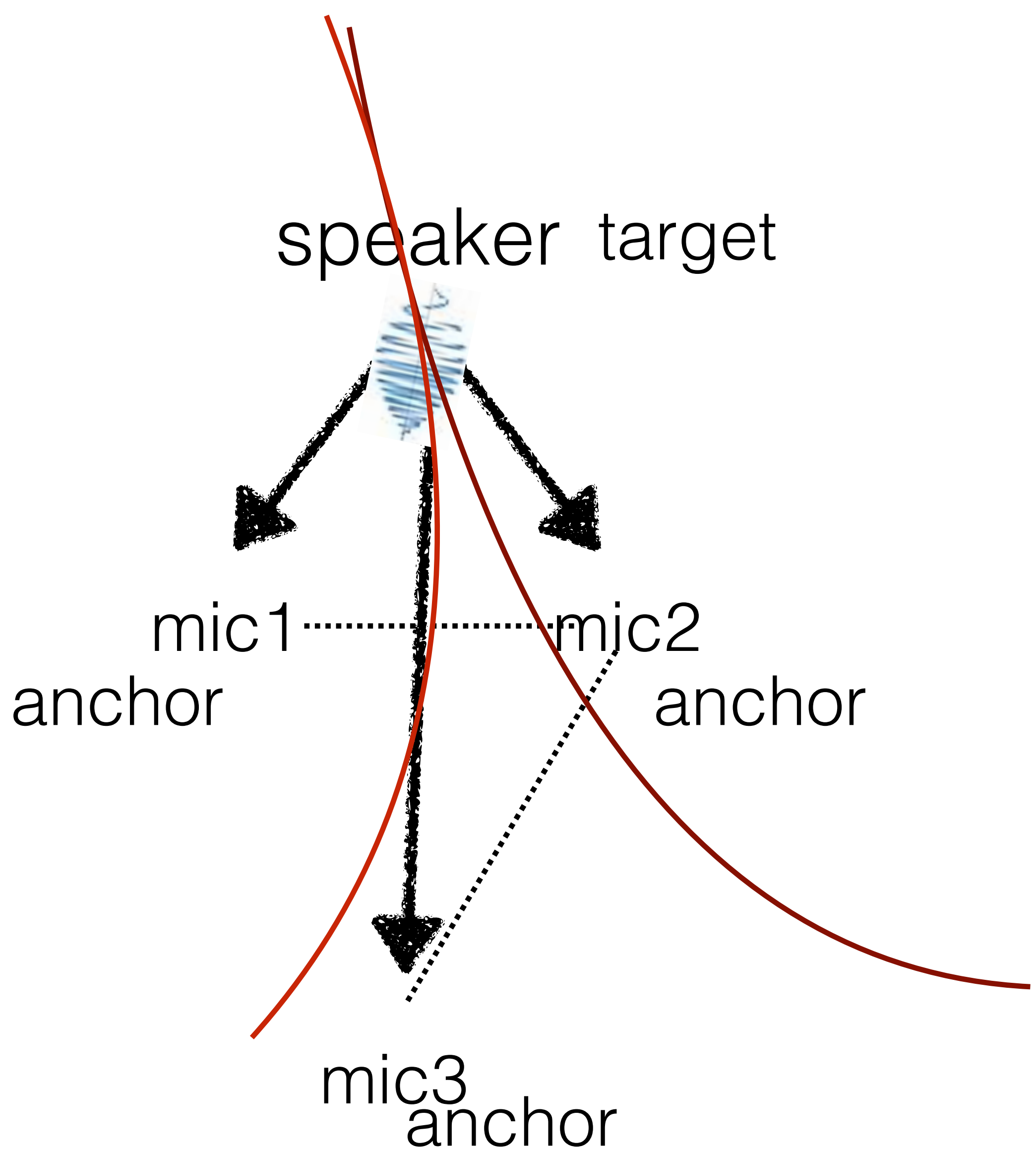


Classic TDoA

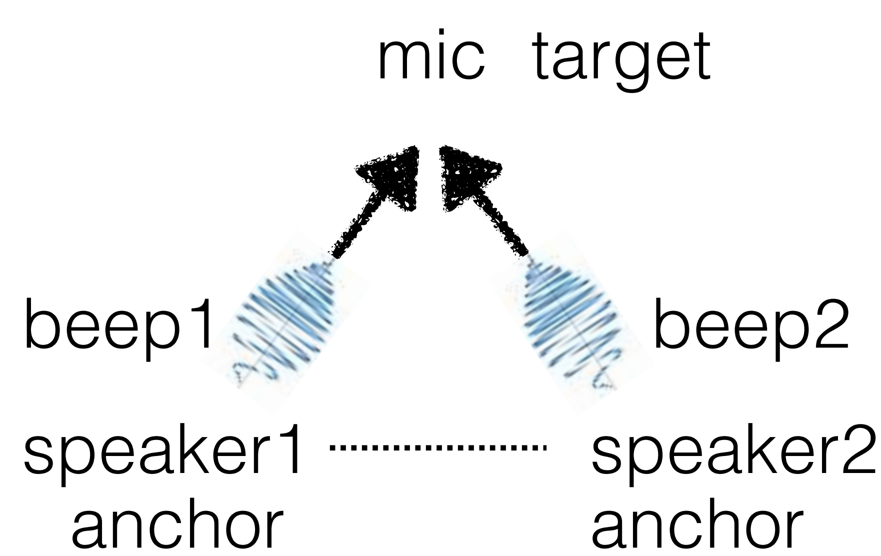


A variant of TDoA

A prime on Time-Difference-of-Arrival

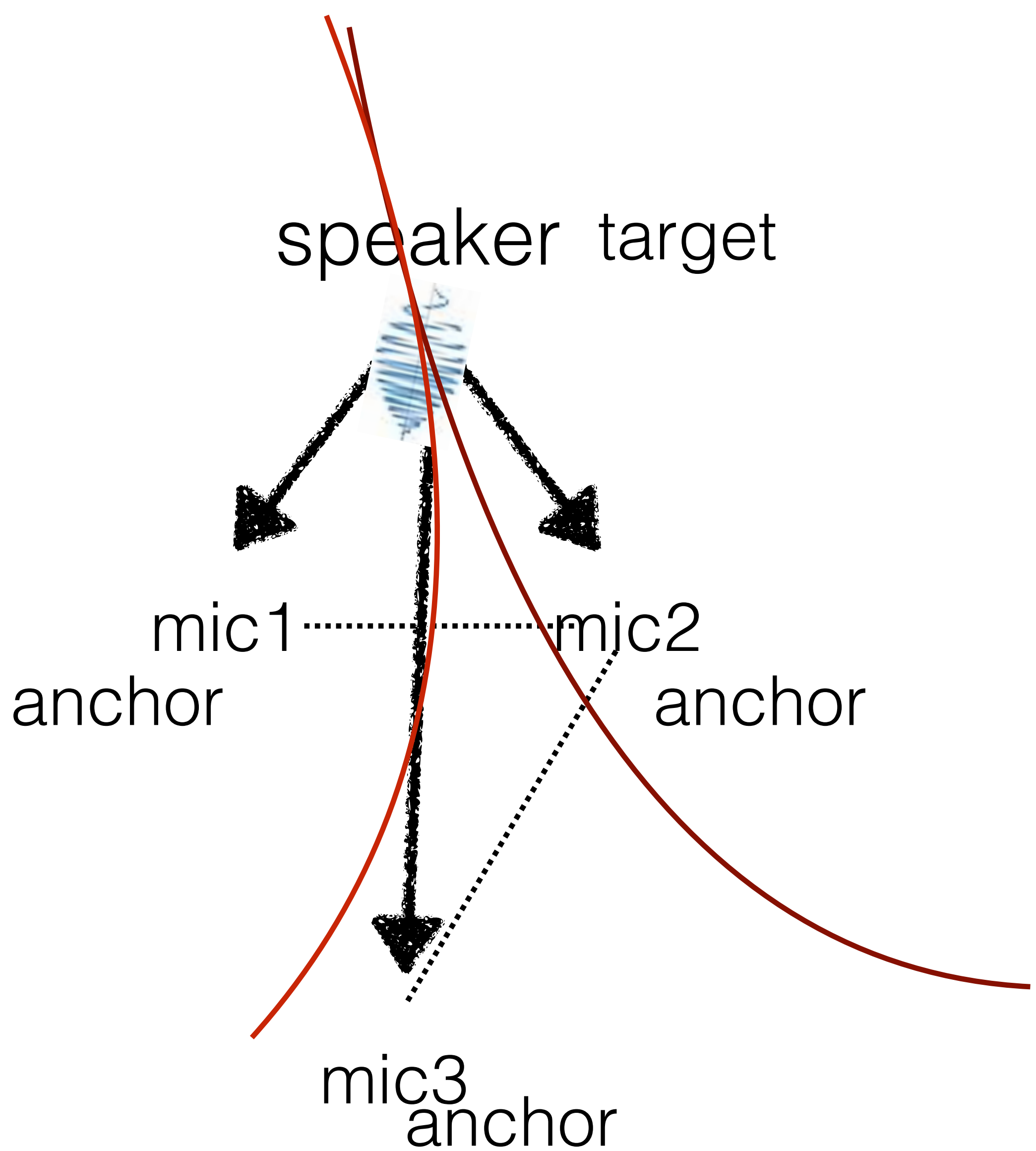


Classic TDoA

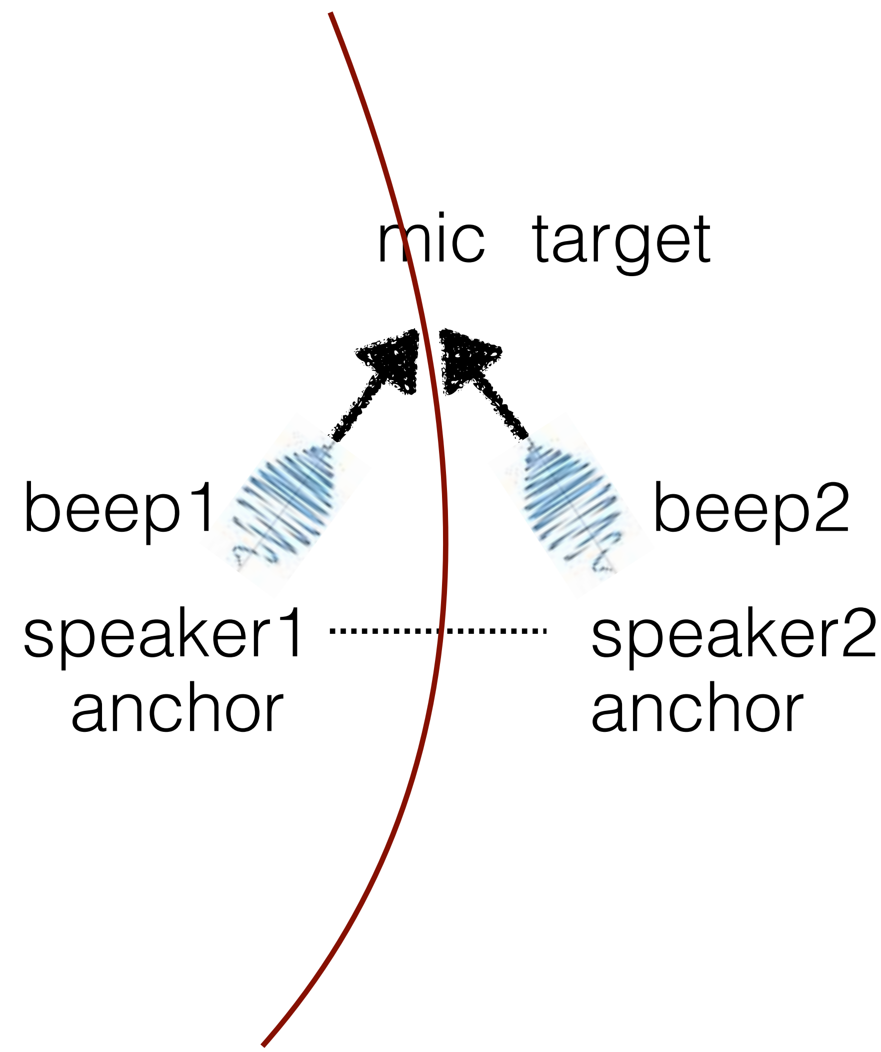


A variant of TDoA

A prime on Time-Difference-of-Arrival

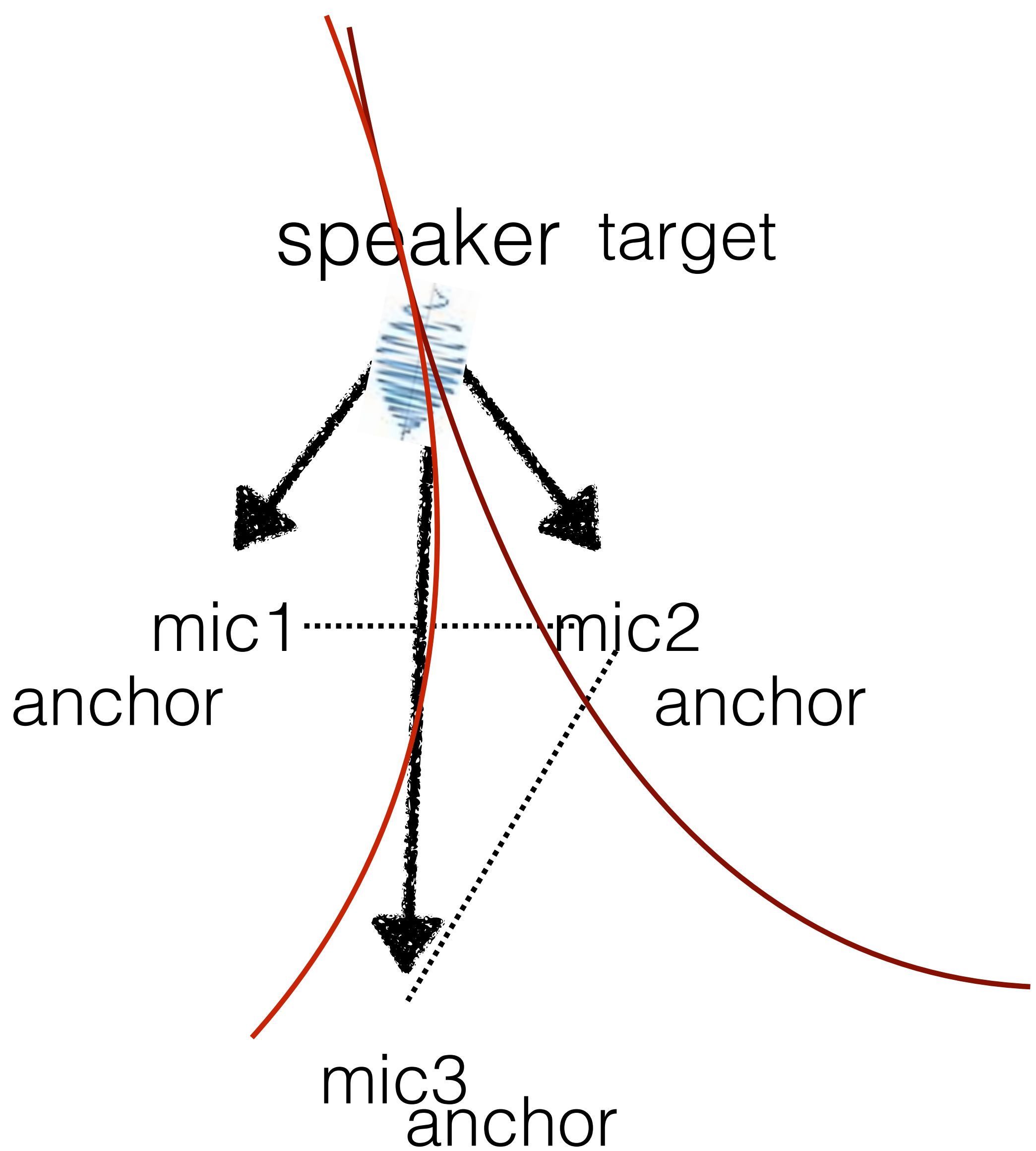


Classic TDoA

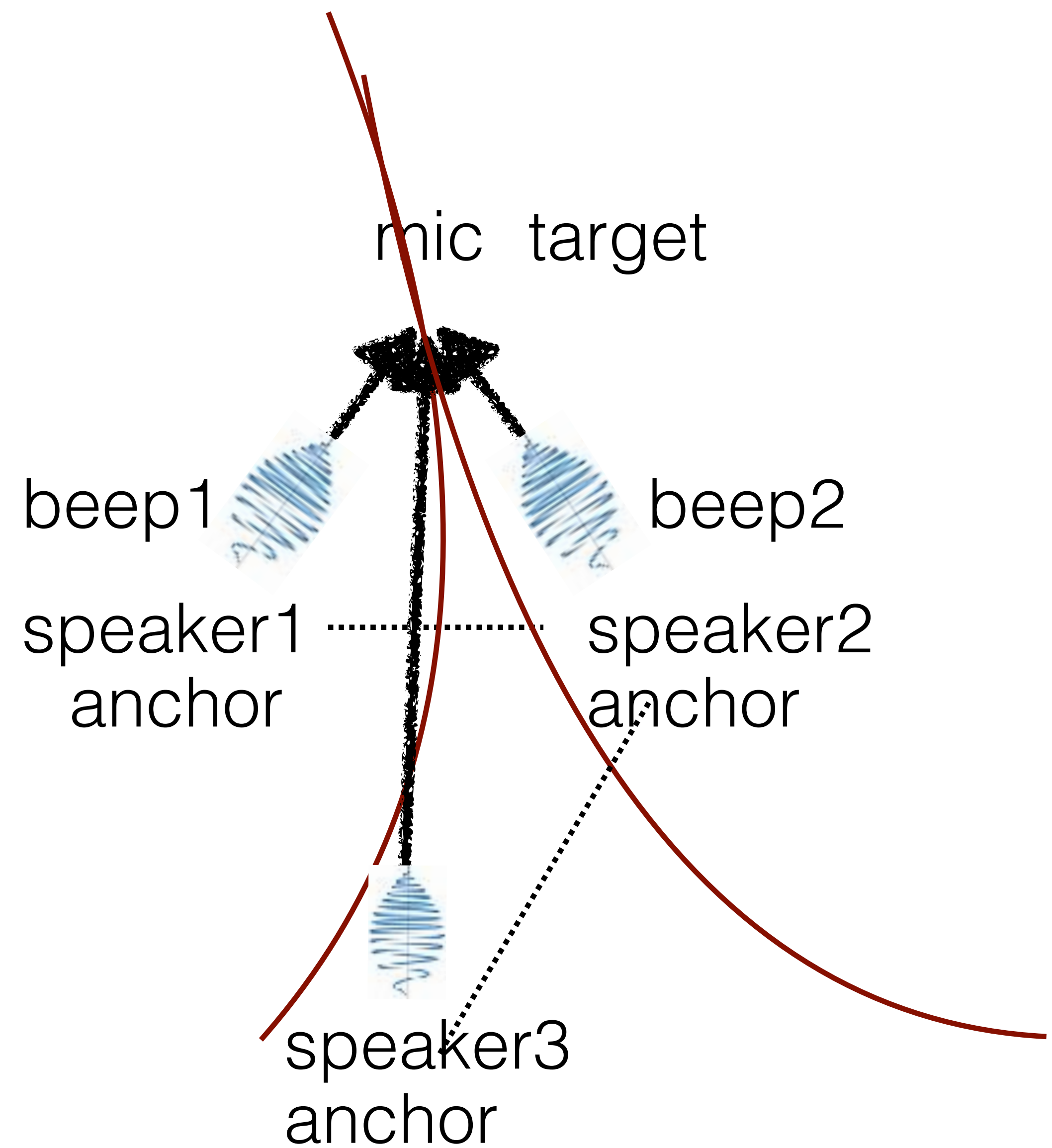


A variant of TDoA

A prime on Time-Difference-of-Arrival

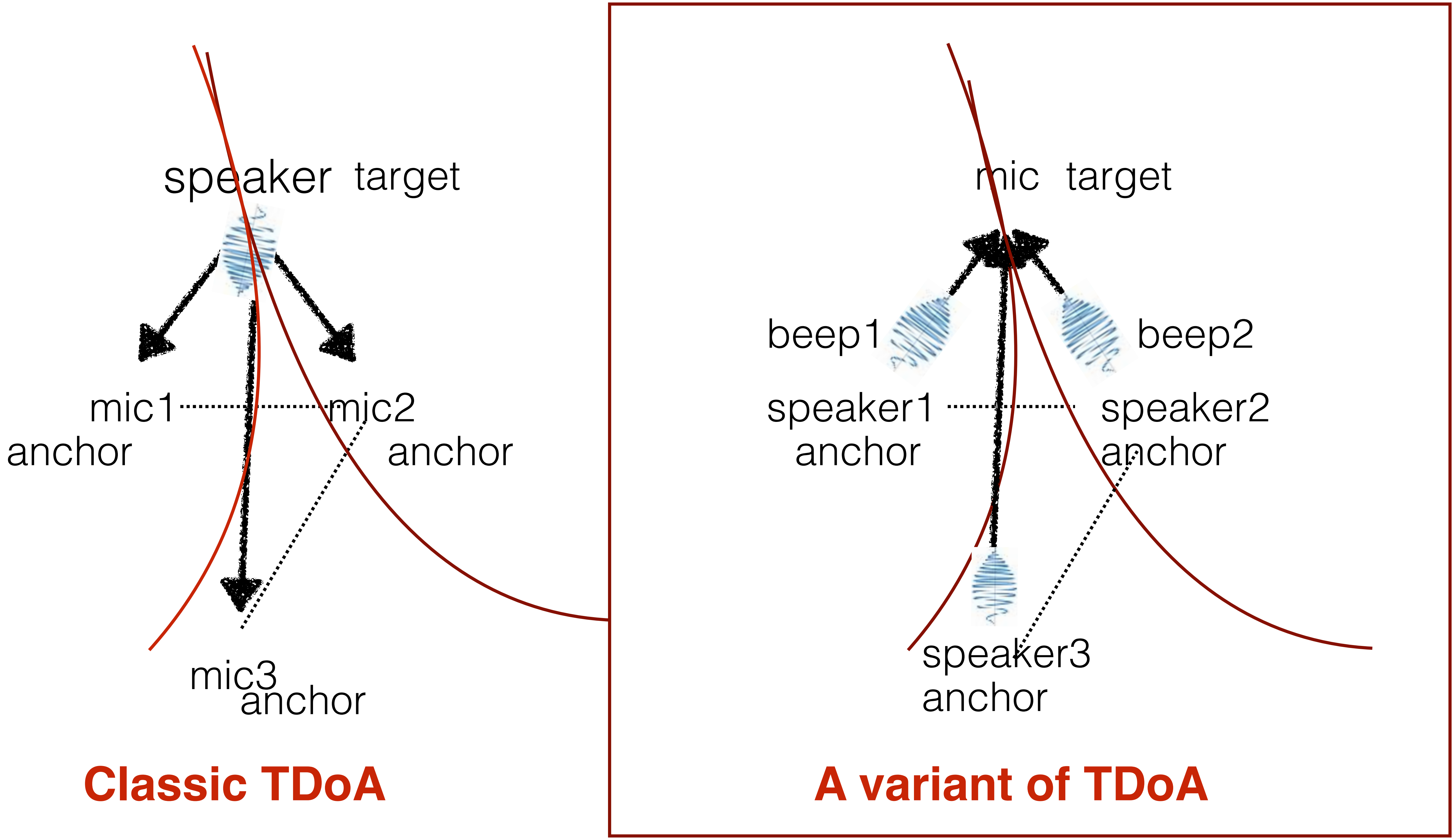


Classic TDoA



A variant of TDoA

A prime on Time-Difference-of-Arrival



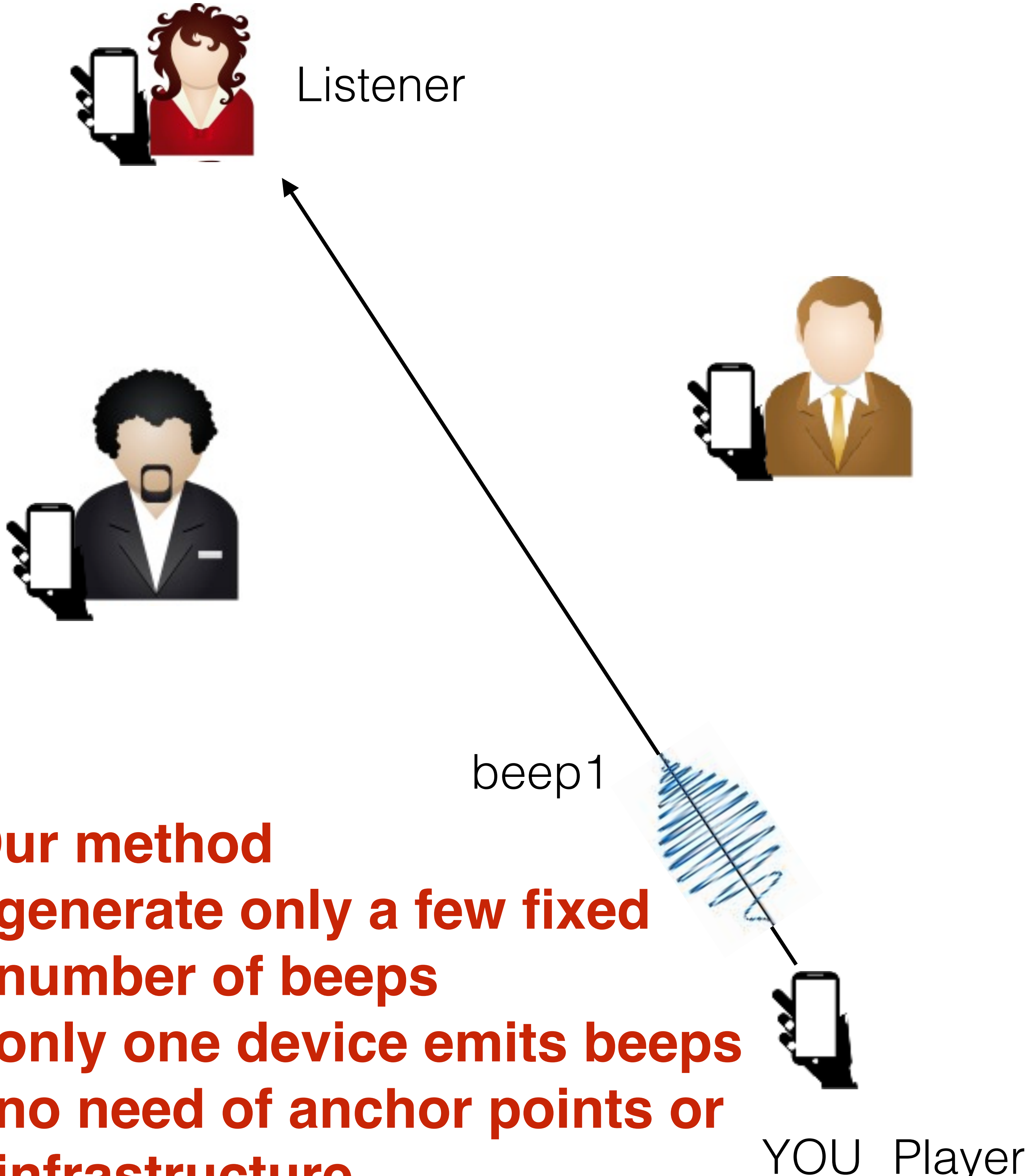
As long as the elapsed time between beeps can be measured, we don't need to send out beeps simultaneously

Listener

Our method

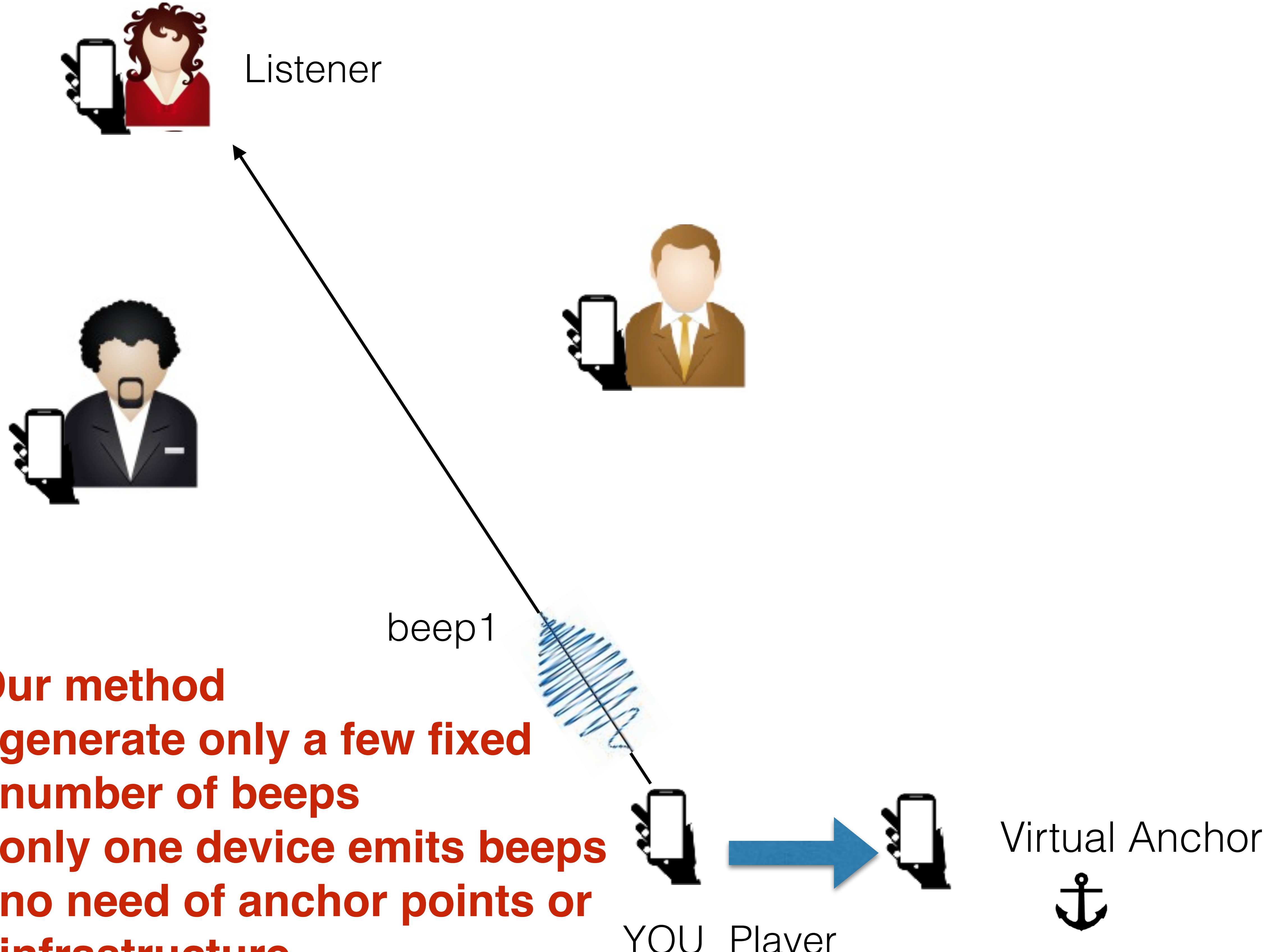
- **generate only a few fixed number of beeps**
- **only one device emits beeps**
- **no need of anchor points or infrastructure**
- **locate all other devices in a single gesture**

YOU Player



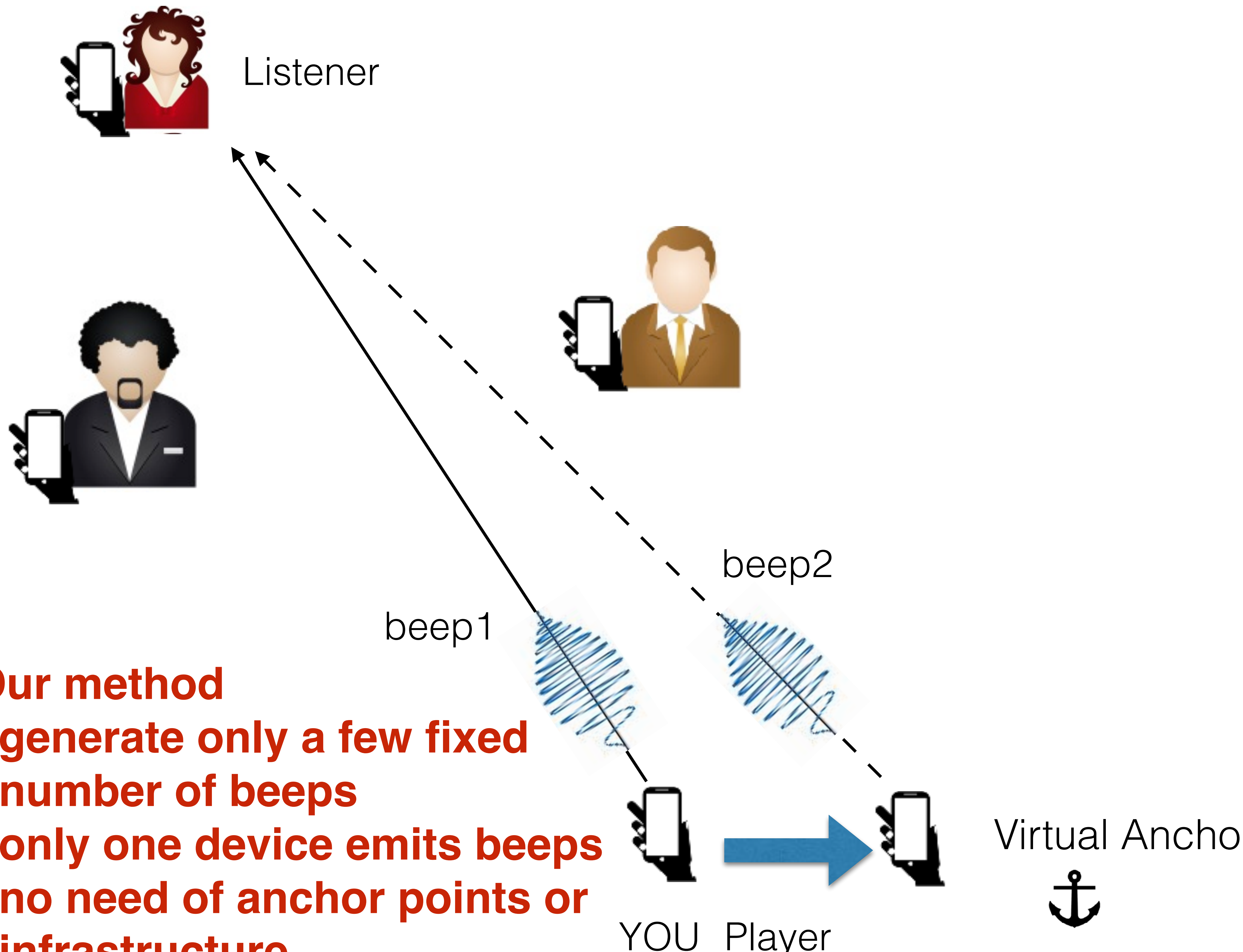
Our method

- **generate only a few fixed number of beeps**
- **only one device emits beeps**
- **no need of anchor points or infrastructure**
- **locate all other devices in a single gesture**



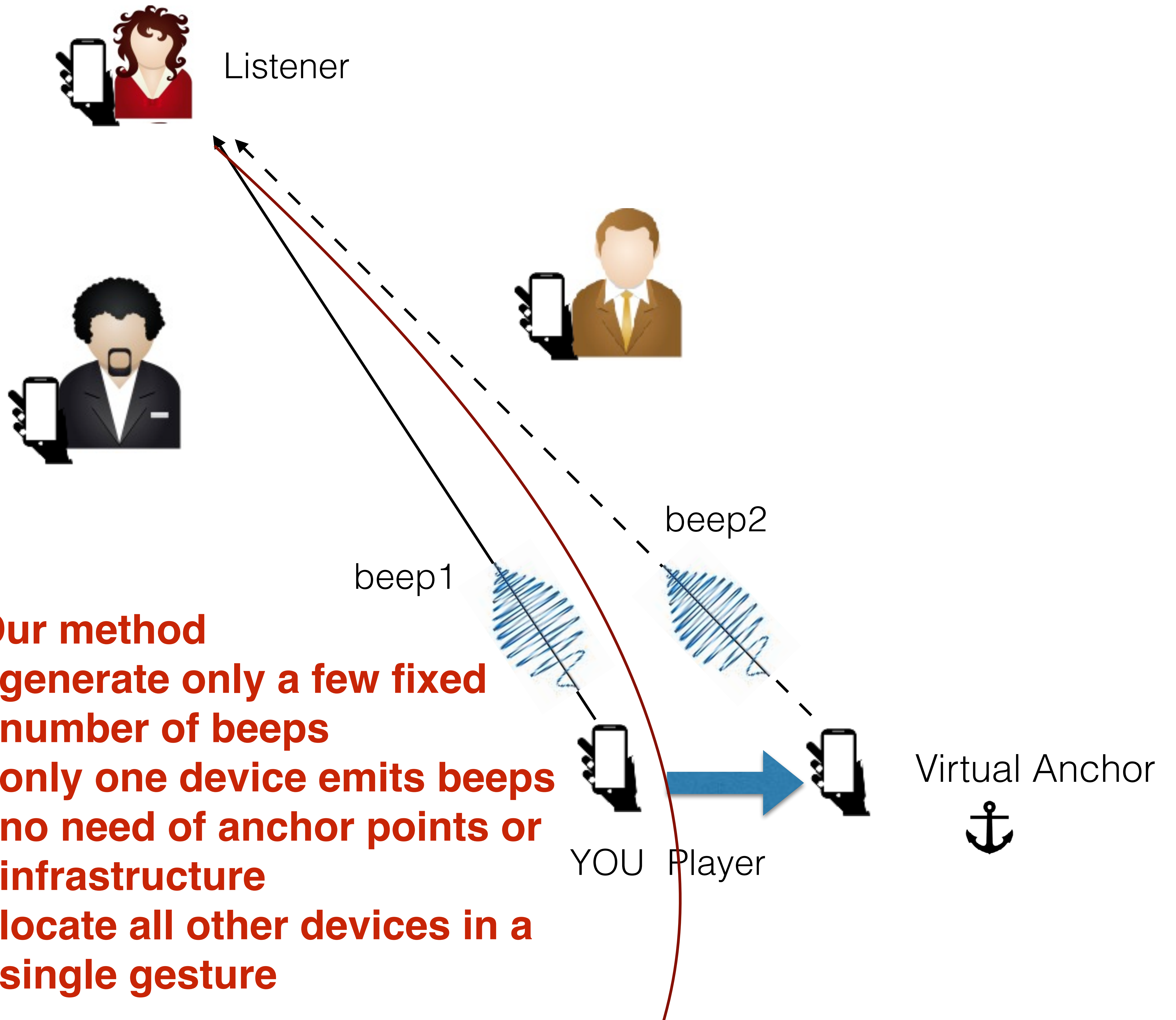
Our method

- **generate only a few fixed number of beeps**
- **only one device emits beeps**
- **no need of anchor points or infrastructure**
- **locate all other devices in a single gesture**



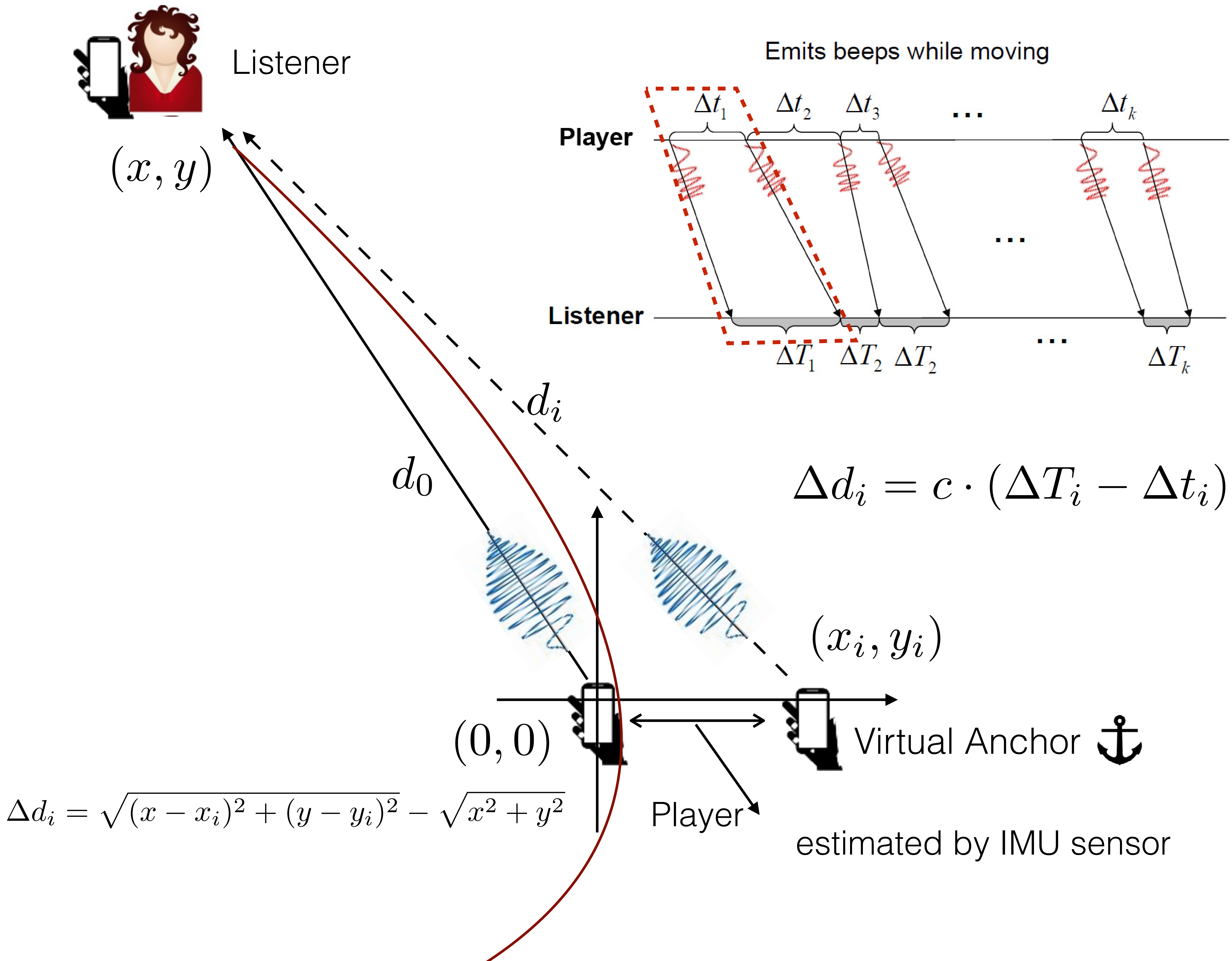
Our method

- **generate only a few fixed number of beeps**
- **only one device emits beeps**
- **no need of anchor points or infrastructure**
- **locate all other devices in a single gesture**



Our method

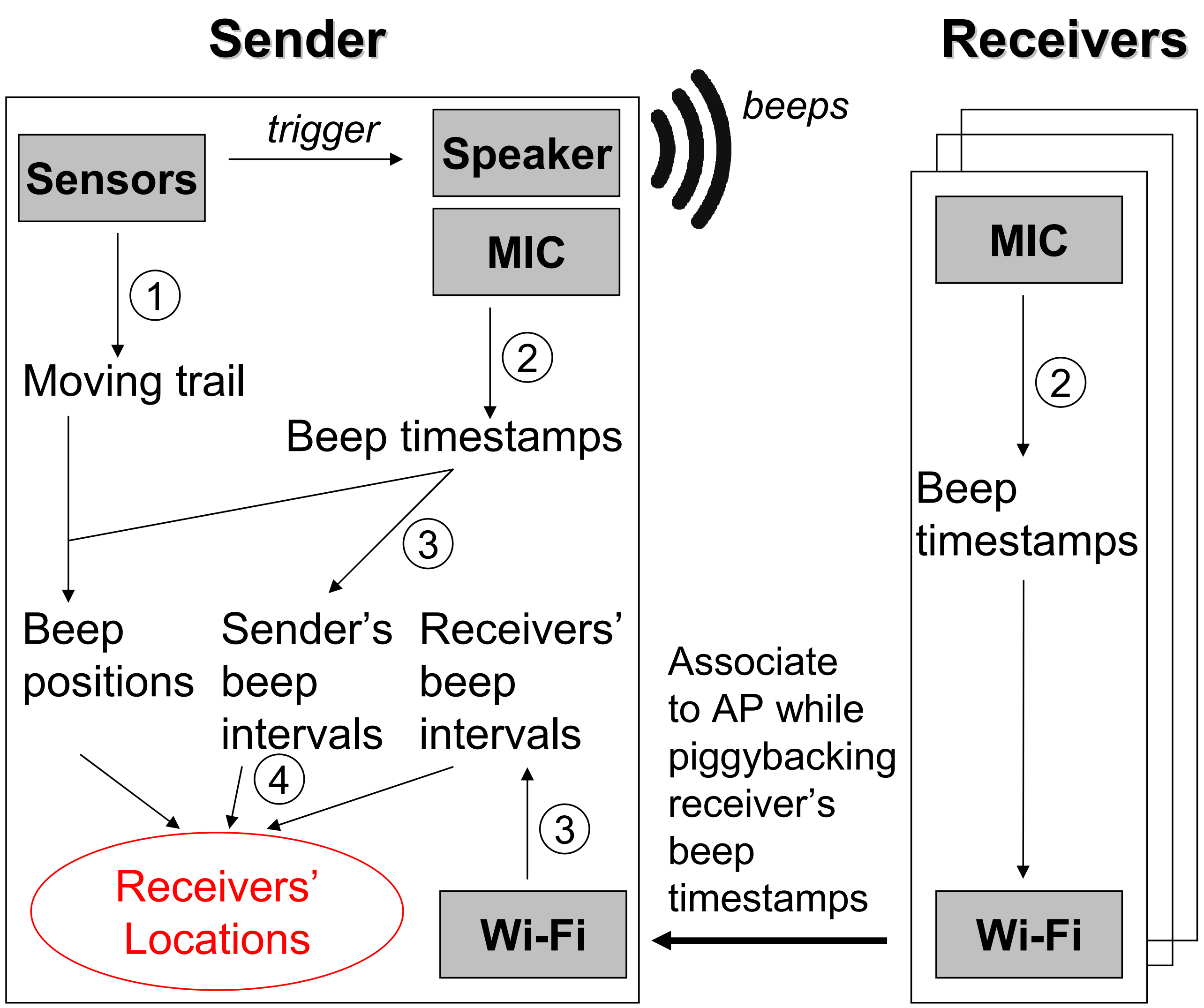
- **generate only a few fixed number of beeps**
- **only one device emits beeps**
- **no need of anchor points or infrastructure**
- **locate all other devices in a single gesture**


Listener
(x, y)
Emits beeps while moving
Δt_1 Δt_2 Δt_3 ... Δt_k
Player
...
Listener
ΔT_1 ΔT_2 ΔT_2 ... ΔT_k
d_i
d_0
$\Delta d_i = c \cdot (\Delta T_i - \Delta t_i)$
(x_i, y_i)
$(0, 0)$
Virtual Anchor
$\Delta d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} - \sqrt{x^2 + y^2}$
Player
estimated by IMU sensor

AMIL: Intuition

- Create virtual anchor by moving the device
  - Utilizing IMU sensors to track the displacement.
- Motion-induced Time-delay of Arrival (TDoA) to localize all other devices in a single run
  - Only the user sends out a fixed number of beeps.
  - all other devices are listeners.

System Architecture



- microphone/speaker
- accelerometer/gyroscope
- three algorithms
  - movement tracking
  - interval calculation
  - positioning

AMIL Algorithms: movement tracking algorithm - Estimate the moving distance

- Accurately estimate the moving displacement of player is crucial to the localization accuracy.

$$\Delta d_i = \boxed{\sqrt{(x-x_i)^2 + (y-y_i)^2}} - \sqrt{x^2+y^2}$$

$$(x_i, y_i)$$

  - double integral: acceleration->speed->distance

- Need to cancel noise, gravity and rotation errors

  - Windowing the accelerometer, using a threshold on the std; only integrate chunks when motion is detected.

  - Using gyroscope to rotation the phone's frame to the initial orientation

    - angle on x,y,z are roll, pitch and yaw ϕ, θ, ψ

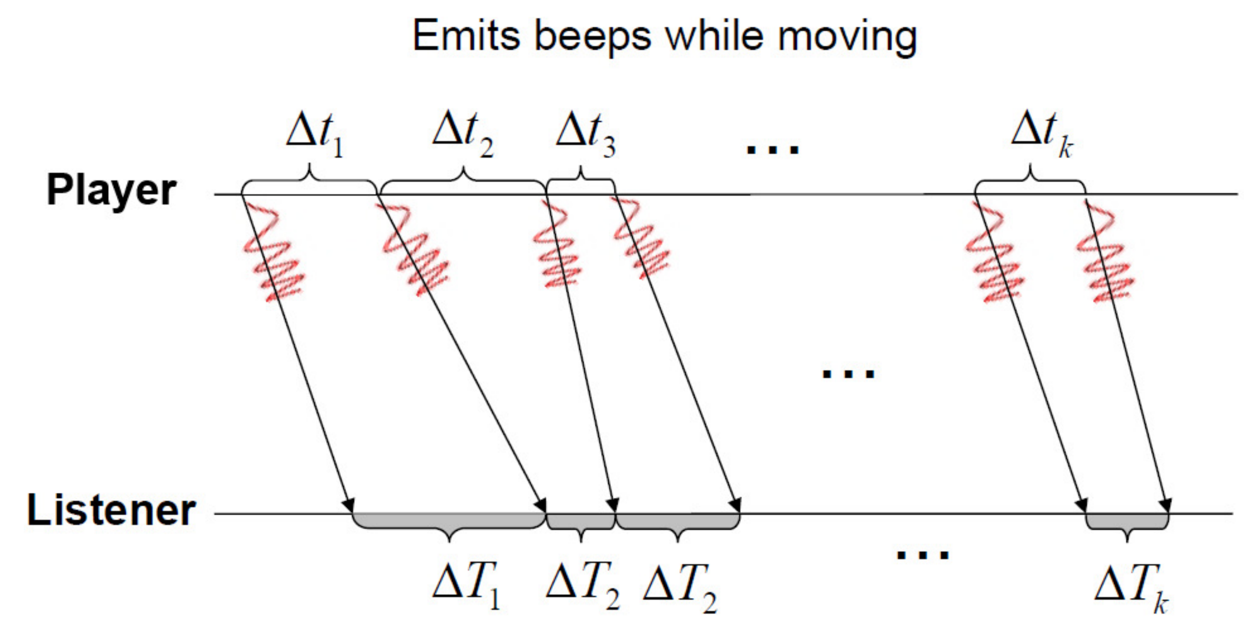
- $$\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} \begin{bmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

AMIL Algorithms: movement tracking algorithm - Cont'd

- To further reduce the error in displacement estimation, we leverage pauses in movements to adjust the velocity.
  - fact: in a series of drawings (e.g. triangle) there are natural pauses
  - the velocities are non-zero due to acceleration residues caused by gravity and misalignment of sensors
  - use a linear method to compensate the velocity of the last movement

$$v'(t_j) = v(t_j) - v(t_k)\frac{t_j - t_i}{t_k - t_i},\ j \in [i, k]$$

AMIL Algorithms: interval calculation algorithm



- determine the delta distance.
  - beep detection:
    - linear chirp,cross-correlation,
    - FFT-based filter.
  - 'self-recording' and 'sample counting' [Beepbeep Sensys'07]:
    - both player and listener records the beep
    - counting audio sample length differences (elapsed time) in two consecutive beeps at both player and listener side
    - the difference of elapsed time at player and listener side can be used to calculate the delta distance.

$$\Delta d_i = c \cdot \boxed{(\Delta T_i - \Delta t_i)}$$

AMIL Algorithms: positioning algorithm

- three beeps

$$\Delta d_1 = \sqrt{(x-x_1)^2 + (y-y_1)^2} - \sqrt{x^2+y^2}$$
$$\Delta d_2 = \sqrt{(x-x_2)^2 + (y-y_2)^2} - \sqrt{x^2+y^2}$$

  - one , two or zeros solutions

- Need more beeps

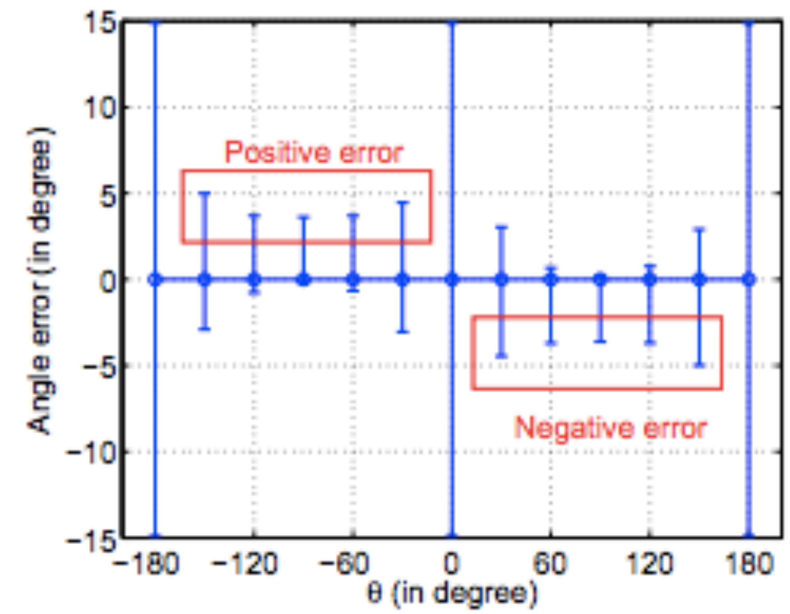
$$\Delta d_i = \sqrt{(x-x_i)^2 + (y-y_i)^2} - \sqrt{x^2+y^2}$$

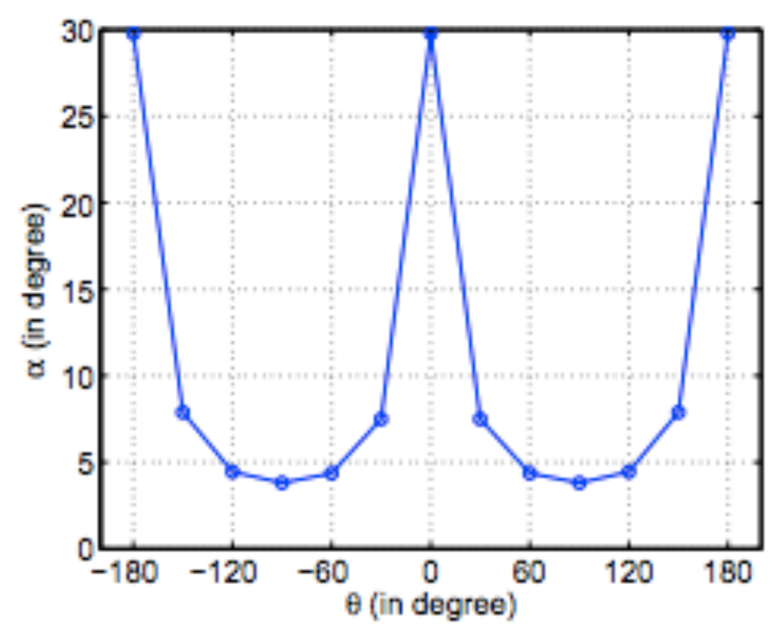
  - select a set of three beeps
  - get all candidates
  - get the centroid

$$\begin{bmatrix} A_1 & B_1 \\ A_2 & B_2 \\ \cdots & \\ A_{n-2} & B_{n-2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ \cdots \\ C_{n-2} \end{bmatrix}.$$

Moving strategy impacts the position accuracy

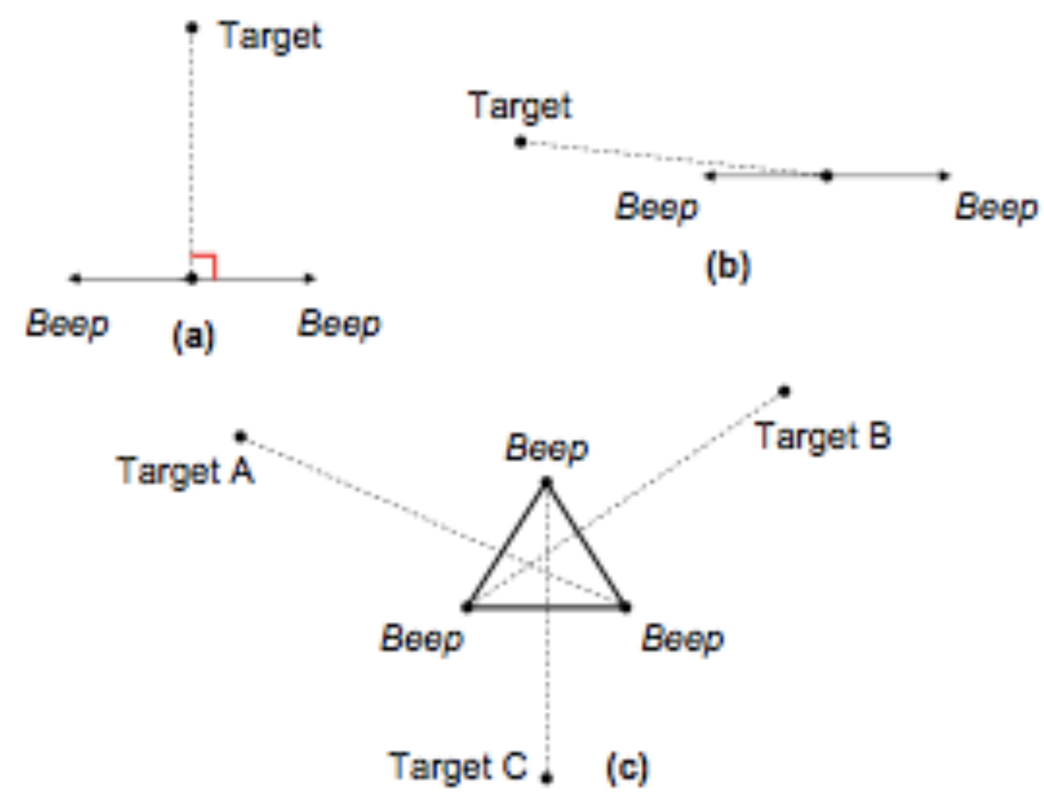
- α as the angle of the sector formed by all position candidates
- θ as the angle between listener and the movement direction
- it is better to move perpendicular than parallel to the direction of the target





Moving strategy impacts the position accuracy - Cont’d

- if the location is known, draw a perpendicular line is the best
- if the location is not known, a triangle would achieve good result on average.



Evaluation

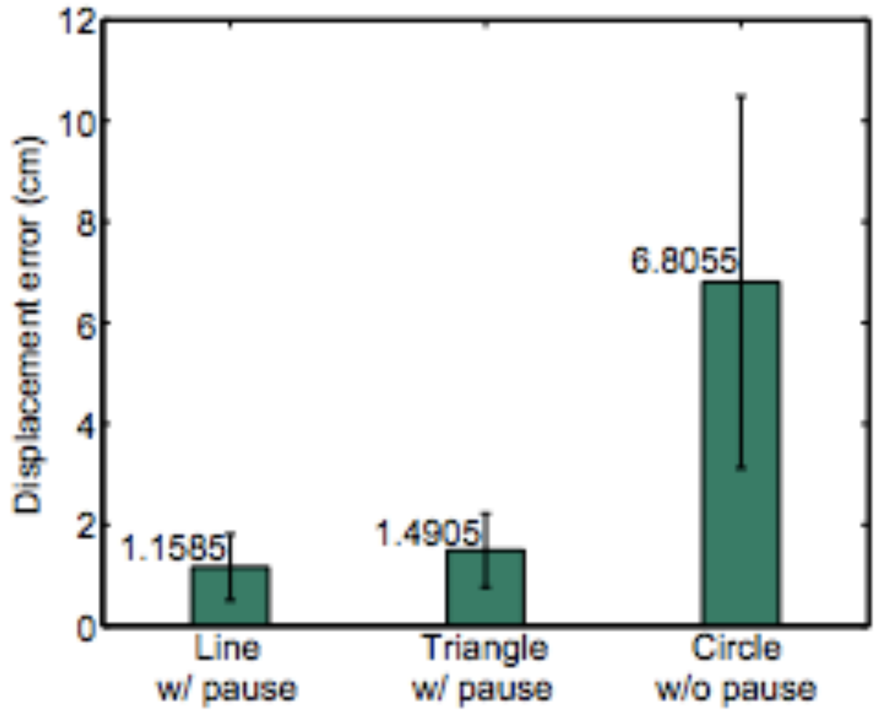
- Galaxy Nexus, Nexus S, Galaxy S2, HTC Evo 3D
- sampling drift compensation
- evaluation questions
  - IMU sensor errors
  - accuracy of direction
  - accuracy of position
  - computation time

Galaxy Nexus (ref)	0.5s	1.0s	1.5s	2.0s	2.5s
Galaxy Nexus	0	0	0	0	0
HTC EVO 3D	3	5	7	9	11
Nexus S	2	3	5	7	9
Galaxy S2	-4	-8	-12	-16	-20

sampling drift under different durations

Evaluation

- Accuracy of IMU sensors

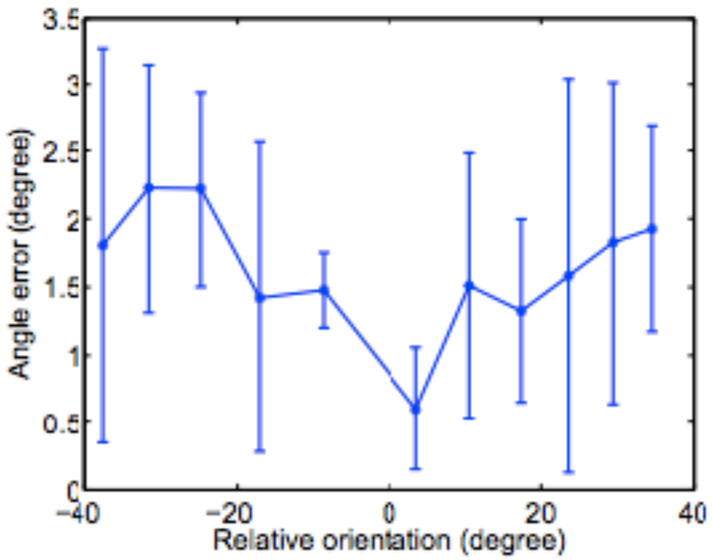


displacement error

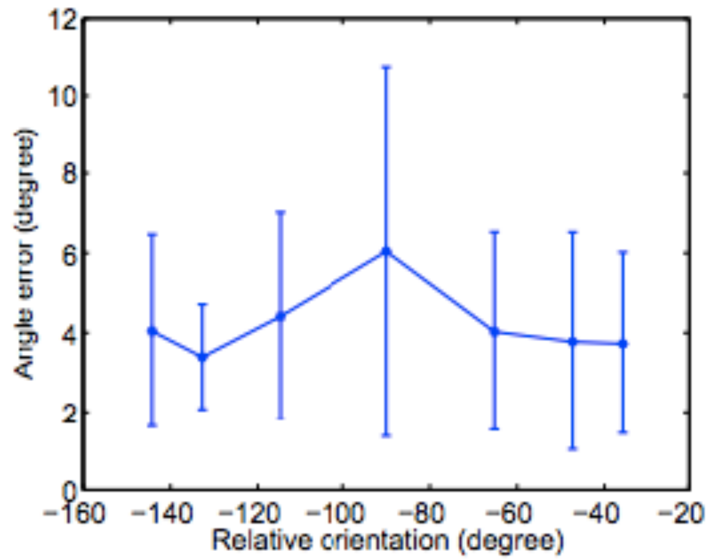
using gestures with pauses can significantly improve the displacement estimation

Evaluation

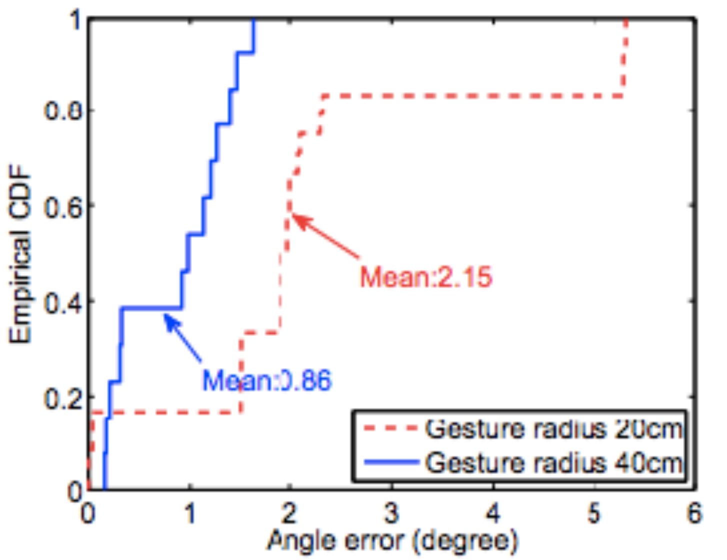
- Accuracy of direction
  - line gesture, triangle gesture ,moving radius



line gesture, 40cm length



triangle gesture, 20cm radius



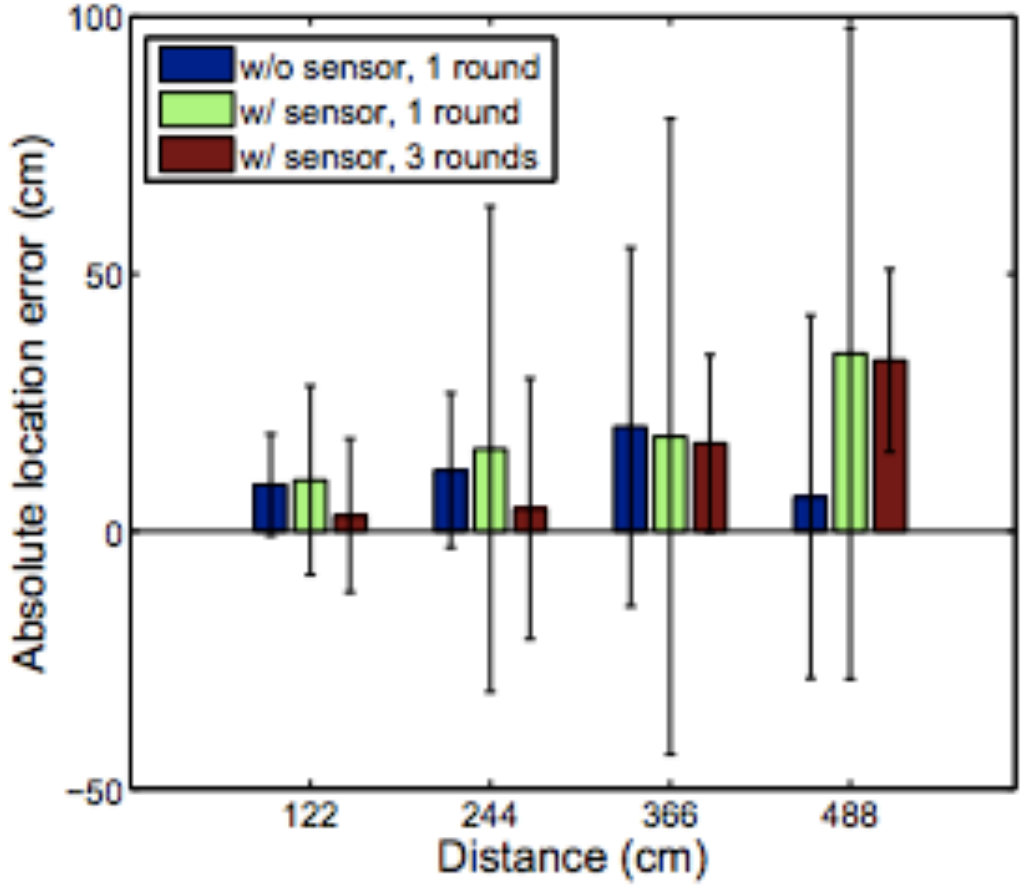
moving radius

average direction errors were within 2.5 ° using a ling gesture, less than 6 ° using a triangle gesture. Larger radius has better accuracy.

Evaluation

- absolute location error

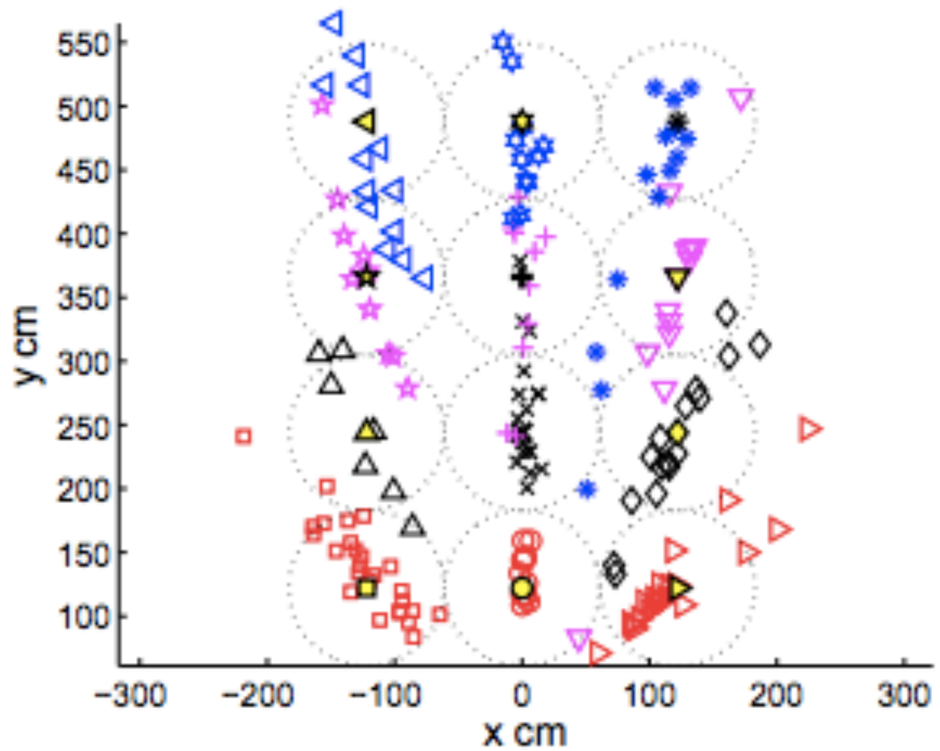
Both error of sensor and beep detection contribute to the localization error. Displacement error add more variations to the location error. By combining three triangles, we can limit the error to less than 50cm for all cases.



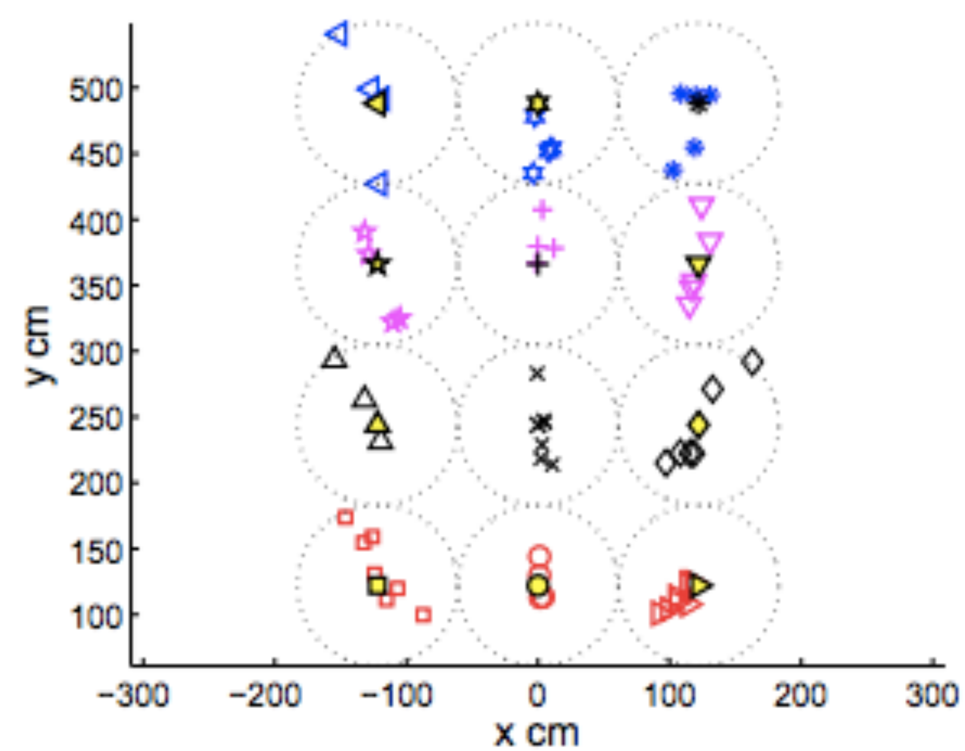
triangle gesture, radius 30.5cm
w/: using sensor to estimate moving distance
w/o: accurate moving distance is supplied

Evaluation

- field test



61 cm radius triangle



61 cm radius, tree triangles

Have good result when target is near the sender. Confirm the accuracy of direction. Multiple round can reduce the error.

Evaluation

- Computation cost

Gesture	Samples		Finishing time (s)			
	Mean	Std	cross-corr. (mean)	cross-corr. (std)	AMIL(mean)	AMIL(std)
Line (4 beeps)	389734	16249	46.06	1.78	3.09	0.07
Triangle (5 beeps)	427008	9523	50.40	1.16	3.68	0.05

FFT-based correlation method can reduce the finish time by more than 90%

Conclusion

- We have investigated the problem grouping and locating mobile phone users in proximity
- AMIL is proposed to leverage a simple gesture to perform localization with motion sensor and speaker/microphone
- We have designed, implemented and evaluated our system on commercial smartphones.
- Extensive experiments have shown that AMIL can achieve less than three degree error in orientation and 50cm error in distance.

End. Thank you.

Q&A