Privacy-Preserving Set Operations

Lea Kissner
leak@cs.cmu.edu

Dawn Song
dawnsong@cmu.edu

- Many practical privacy problems share certain characteristics:
  - Several parties, each with a private input
  - The data cannot be freely shared
  - The parties wish to *privately* compute some function of their joint inputs
  - Often, the inputs are sets or multisets

The Do-Not-Fly List

Airline Flight List

Government Terrorist List

The Do-Not-Fly List

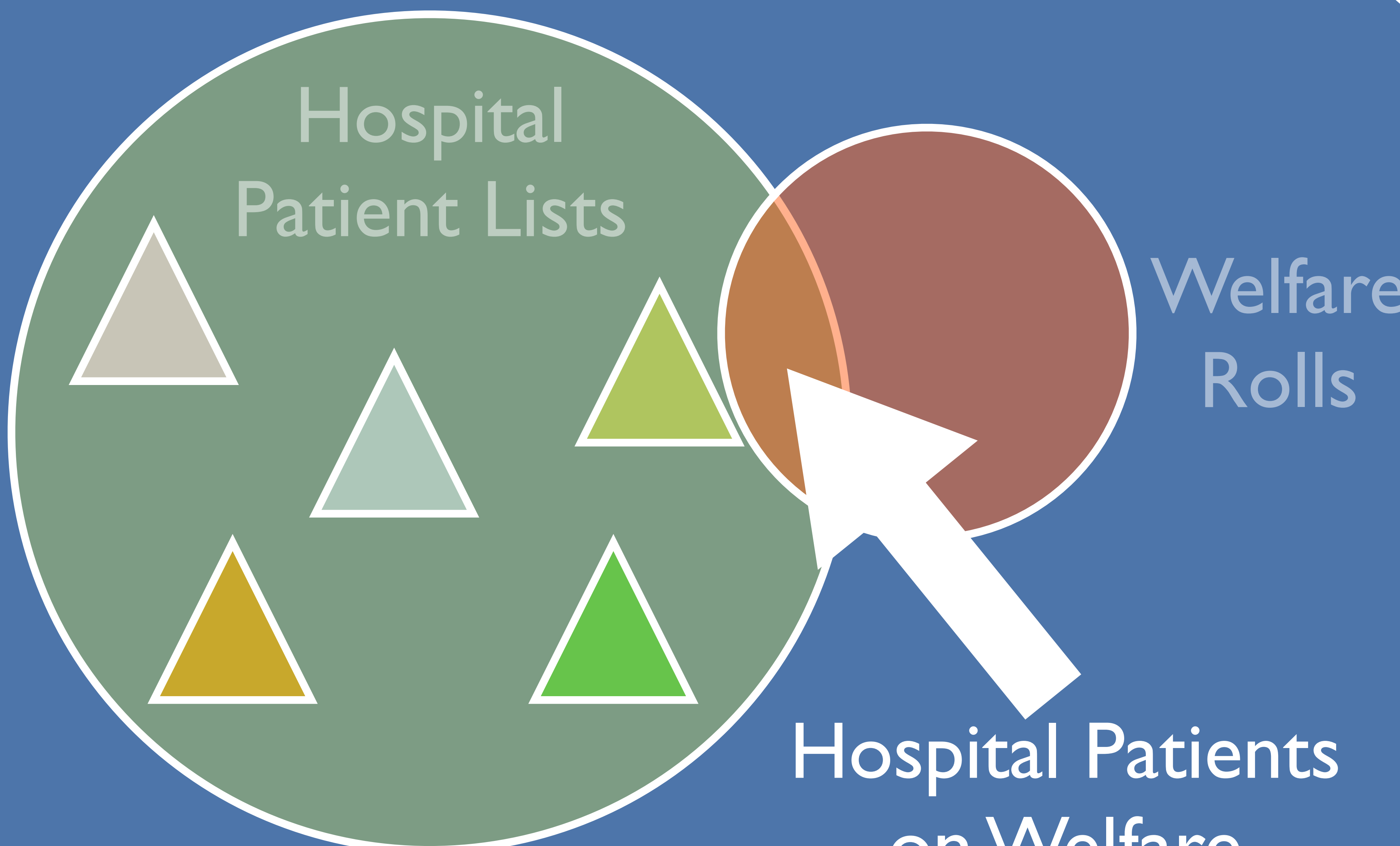
Statistics-Gathering

Hospital
Patient Lists

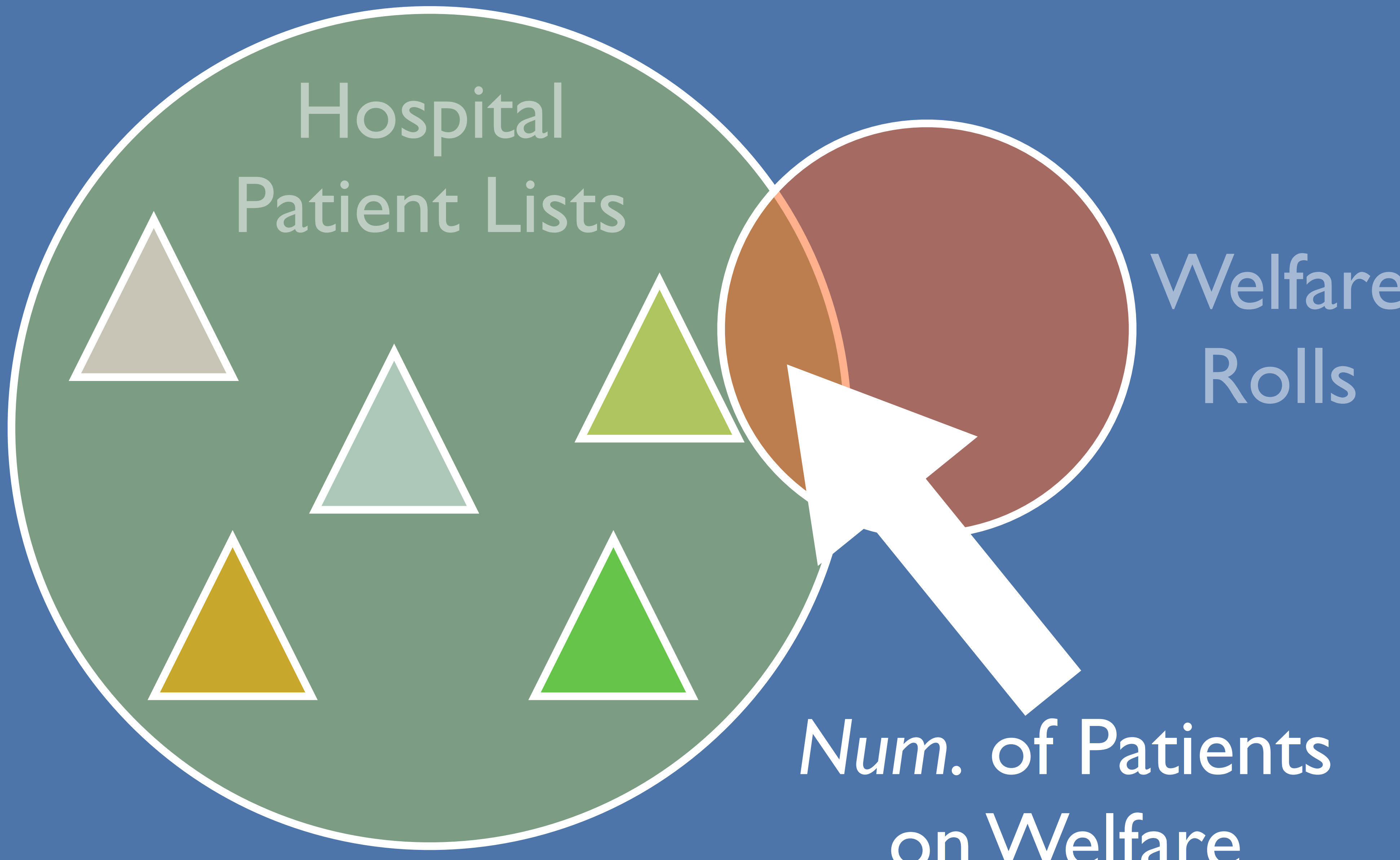
Welfare
Rolls

Statistics-Gathering

Statistics-Gathering



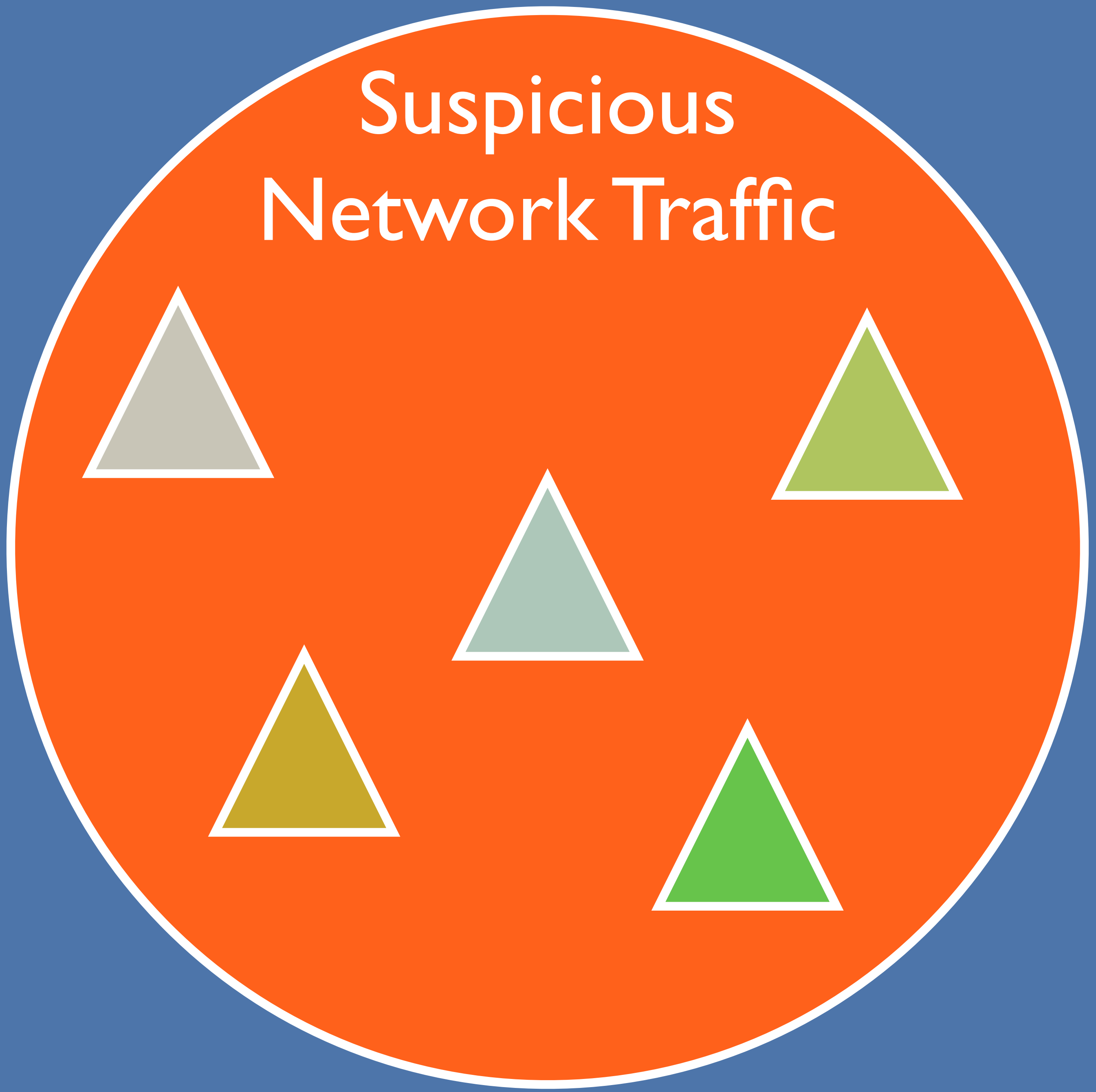
Statistics-Gathering



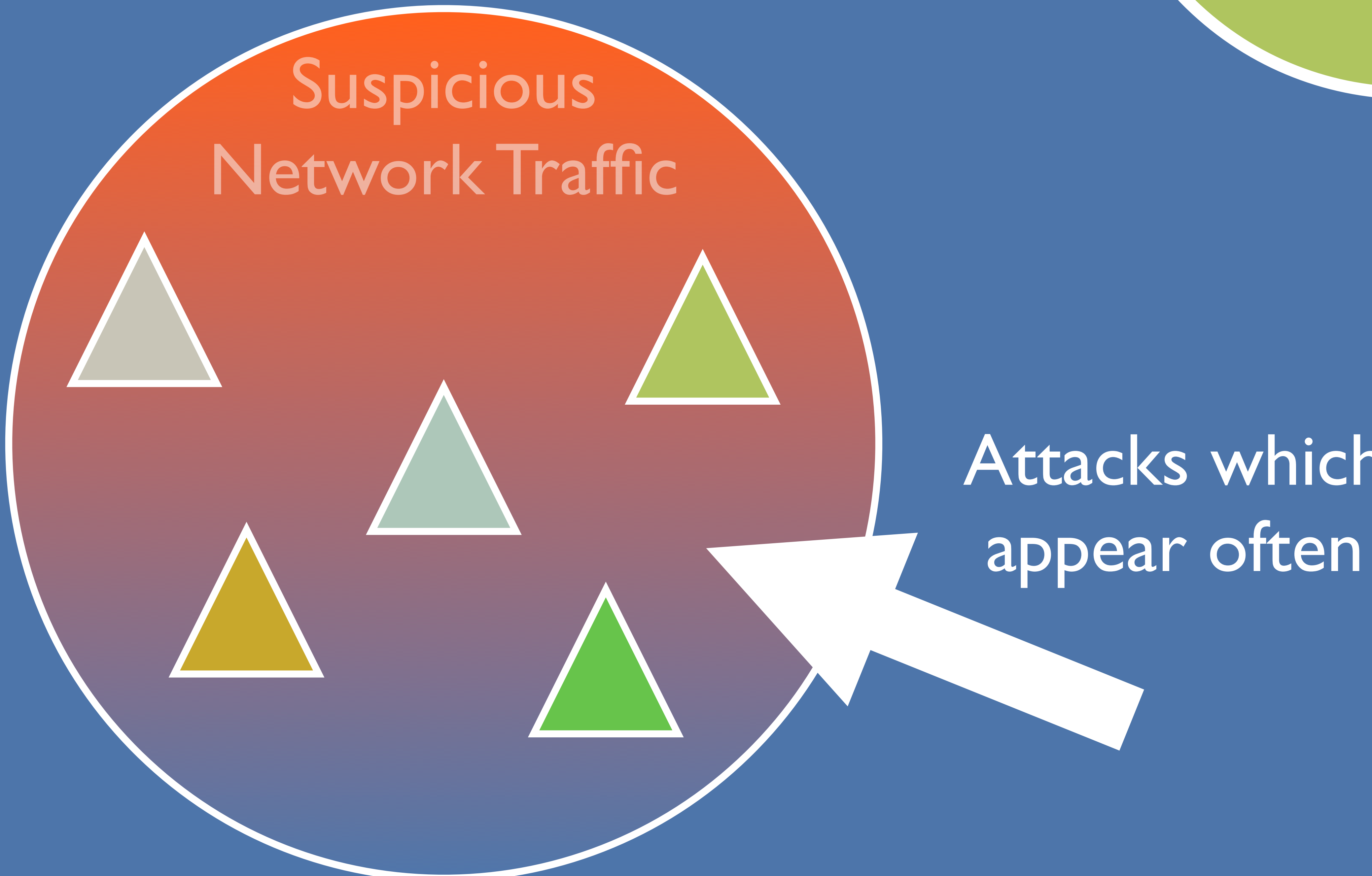
Network Monitoring

Suspicious
Network Traffic

Network Monitoring



Network Monitoring



Prescription Cheaters

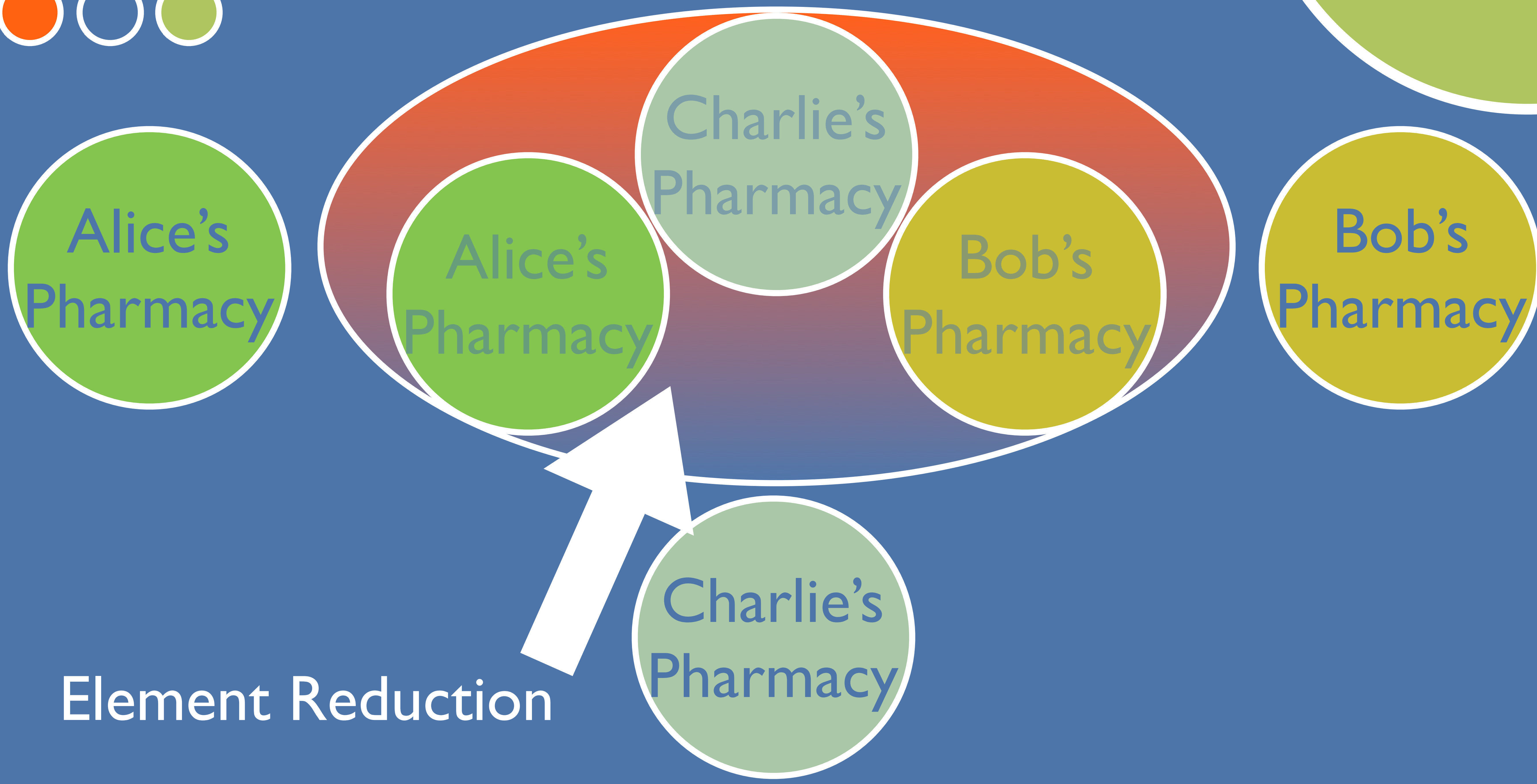
Alice's Pharmacy

Bob's Pharmacy

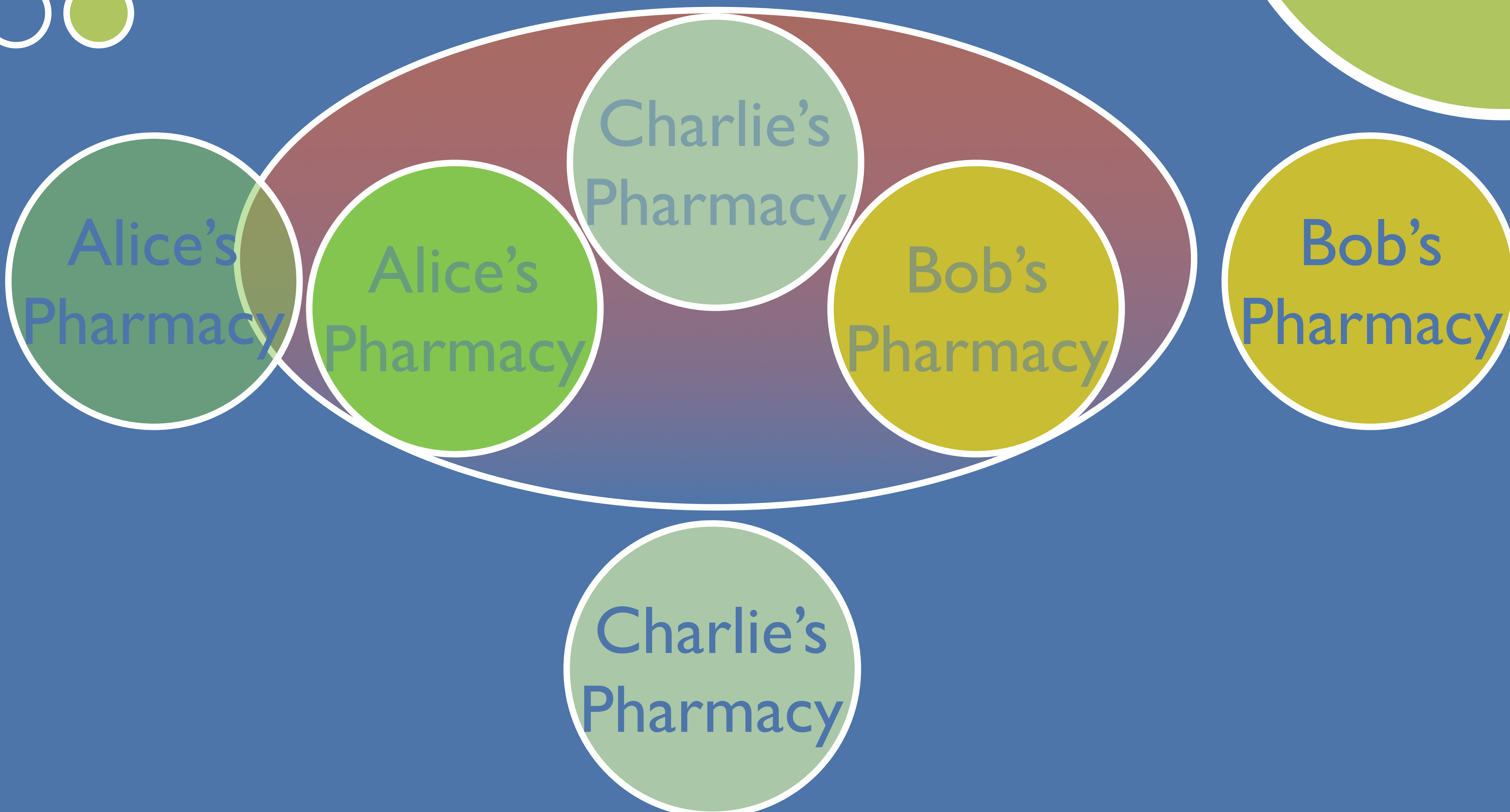
Charlie's Pharmacy

Prescription Cheaters

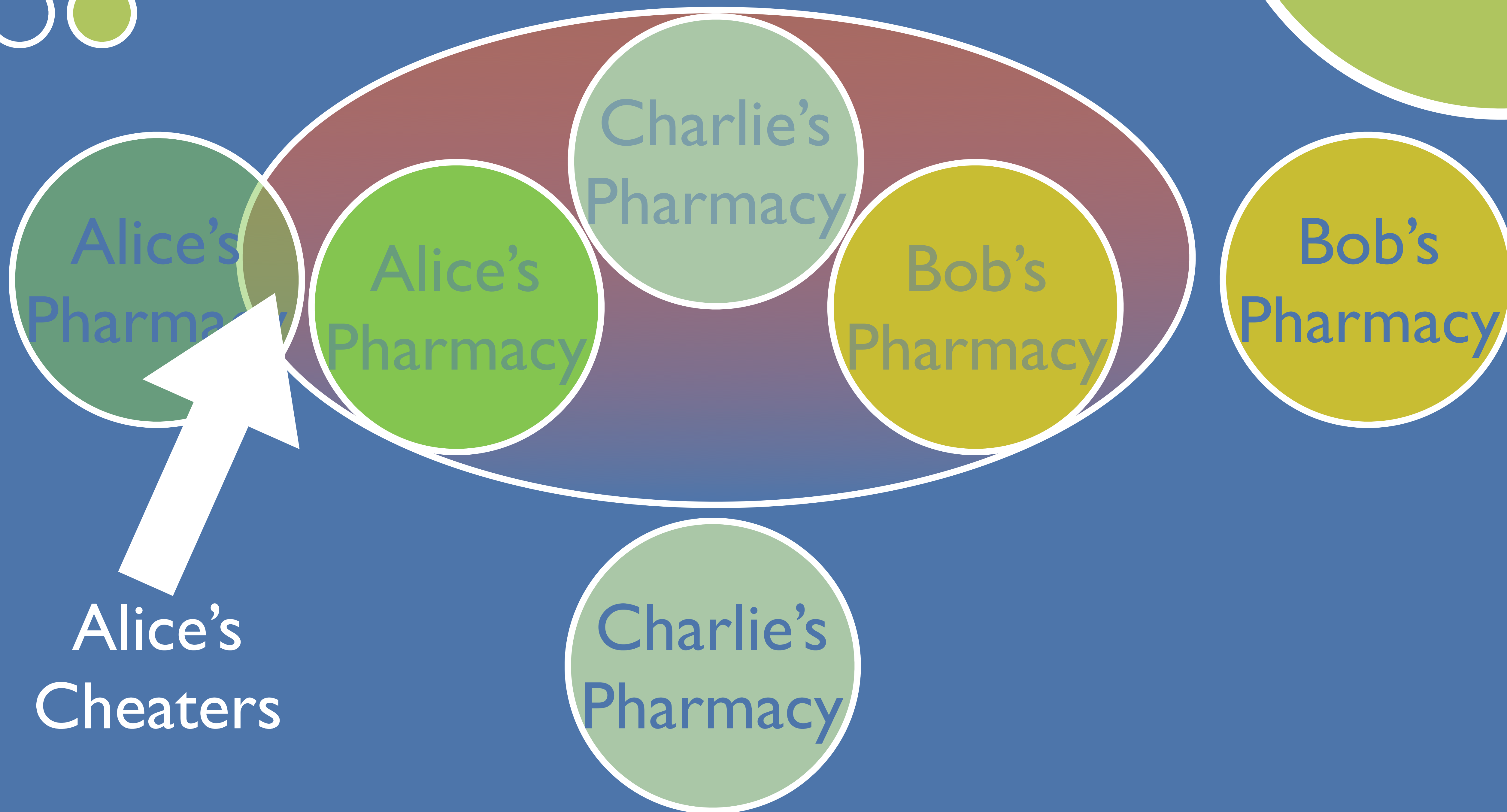
Prescription Cheaters



Prescription Cheaters

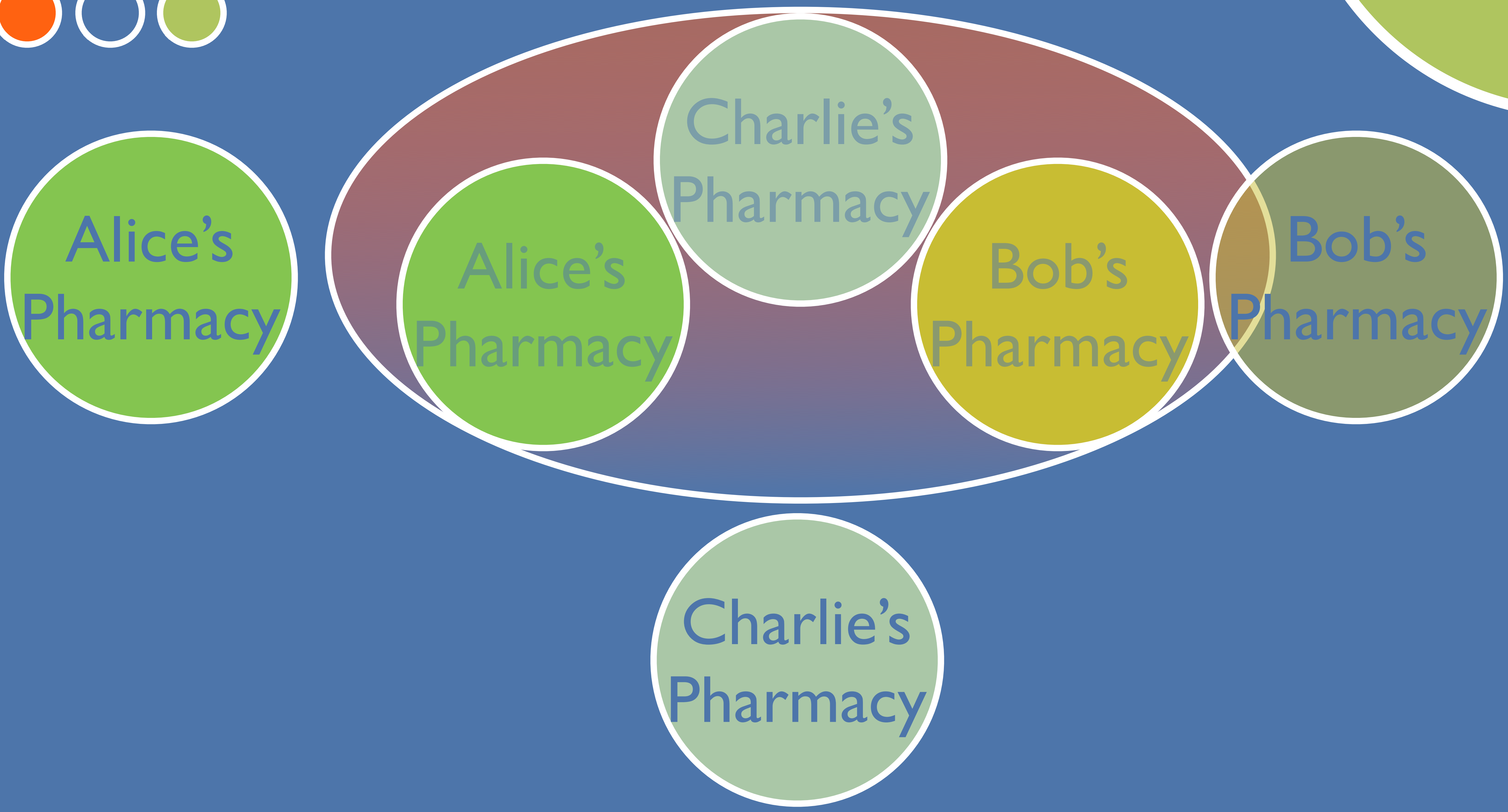


Prescription Cheaters

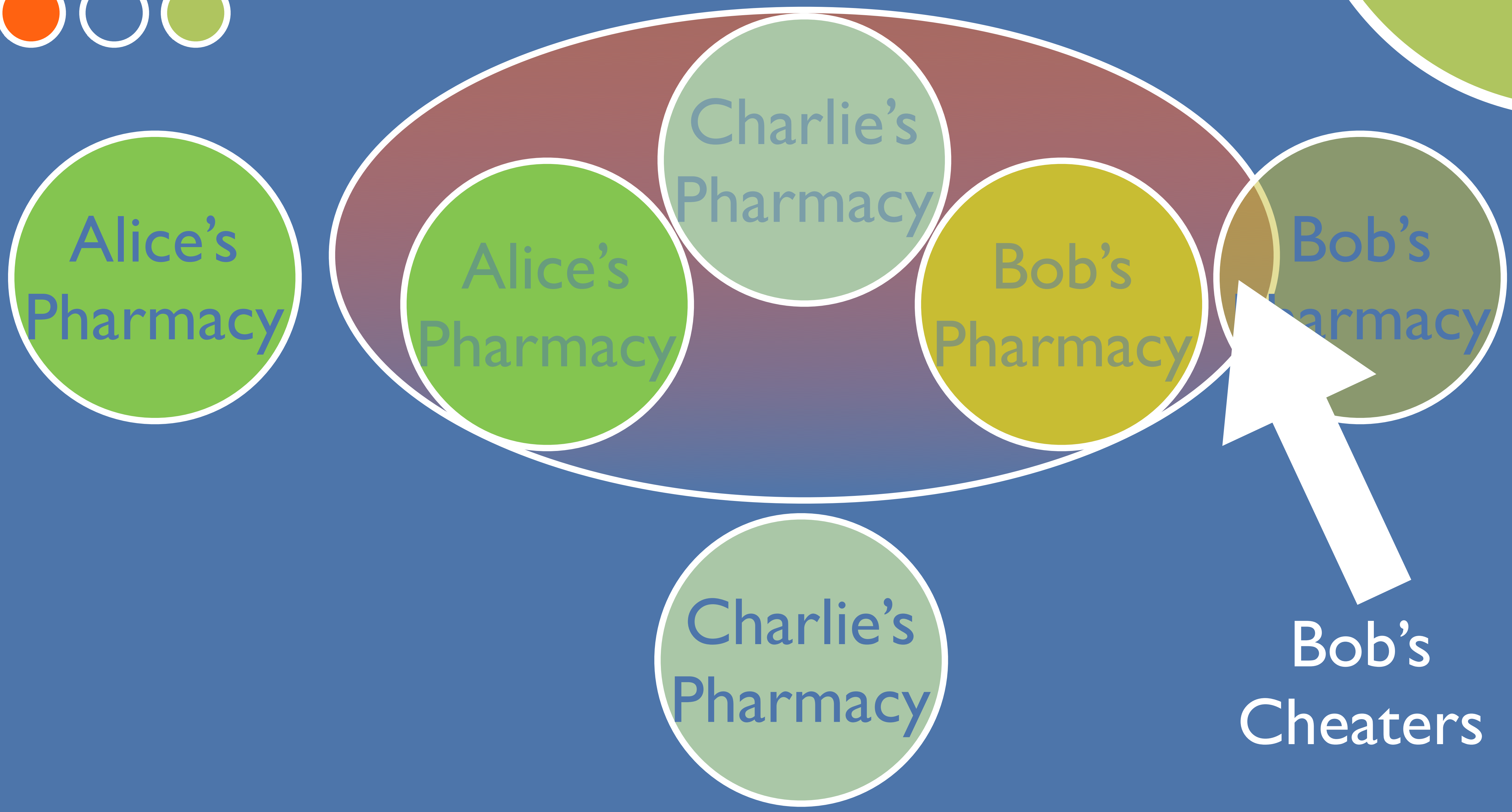


Prescription Cheaters

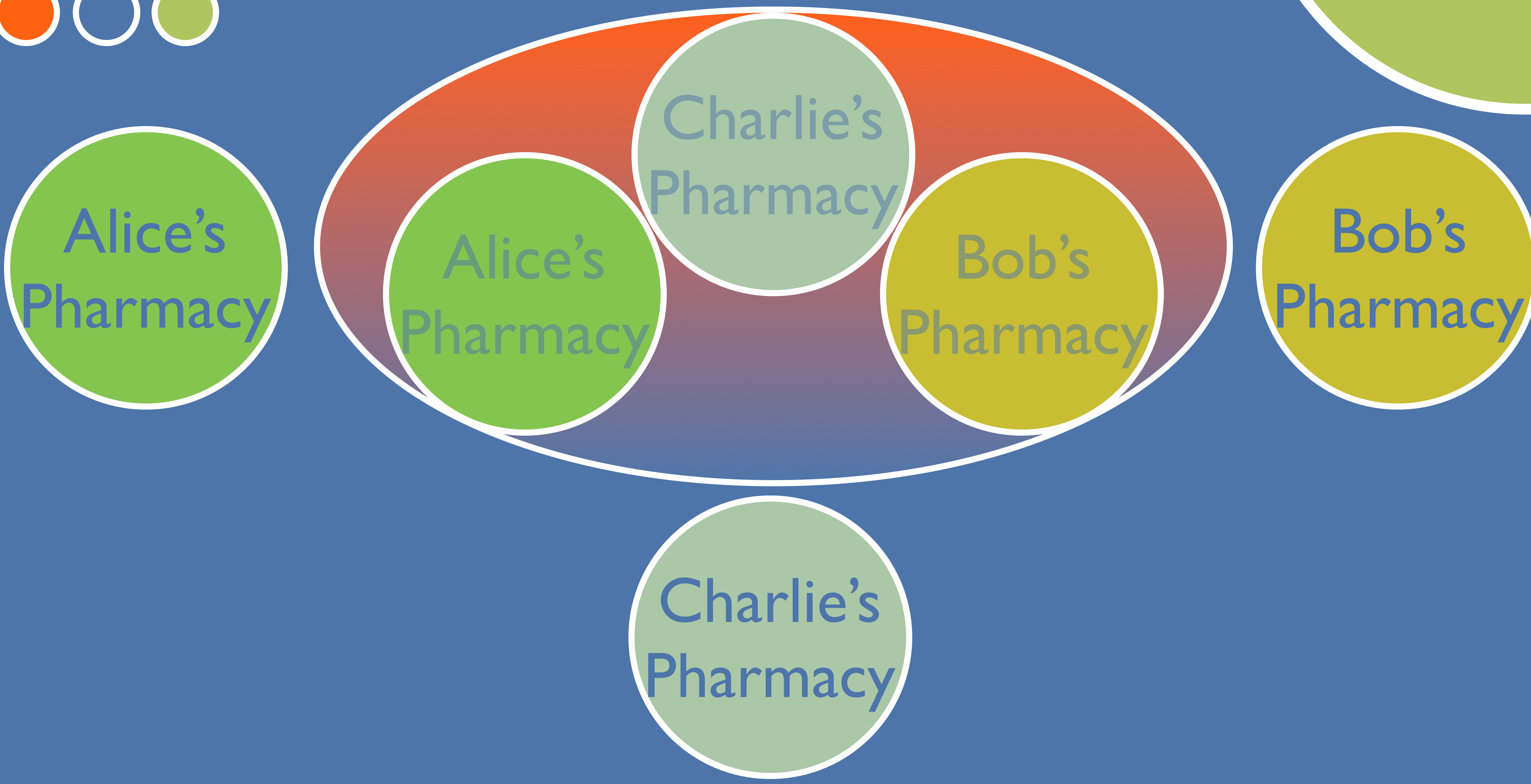
Prescription Cheaters



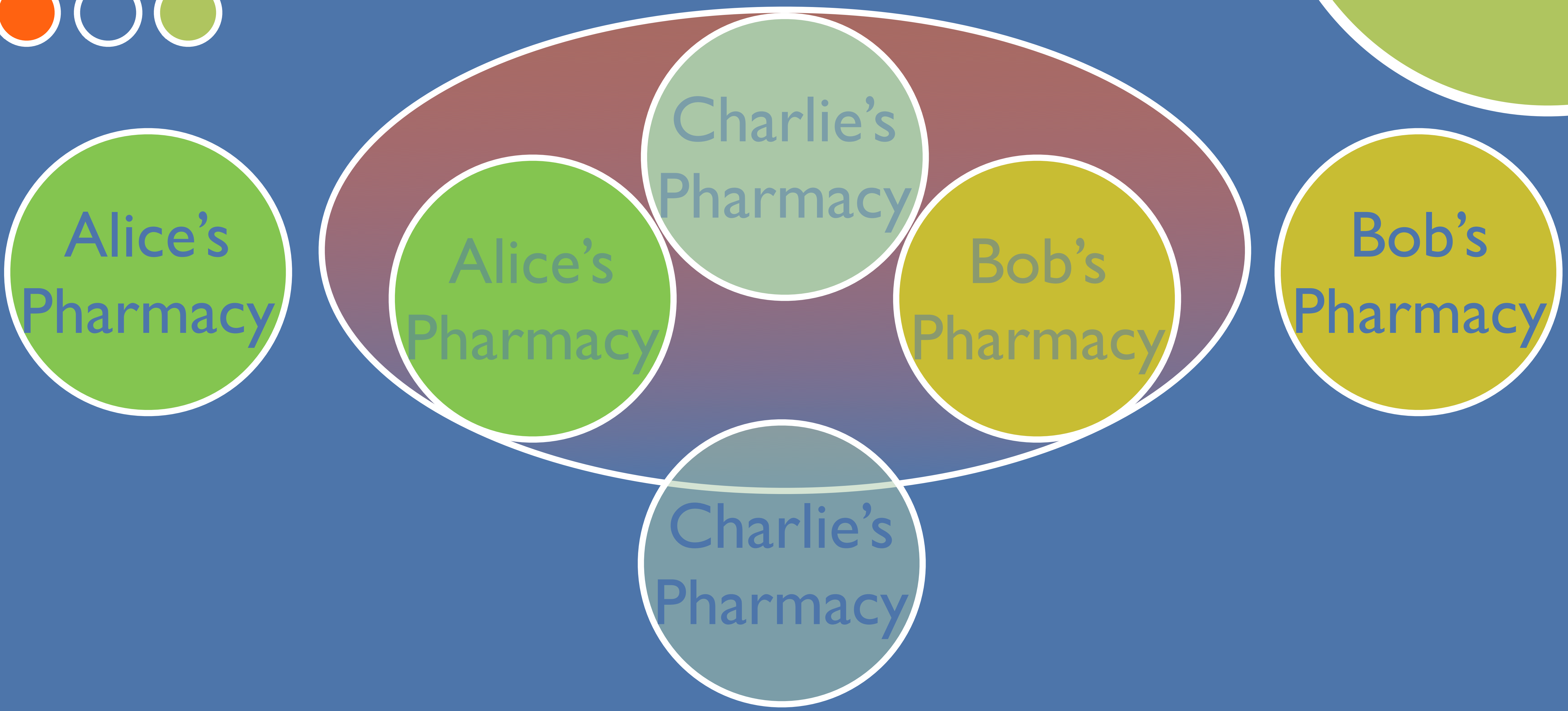
Prescription Cheaters



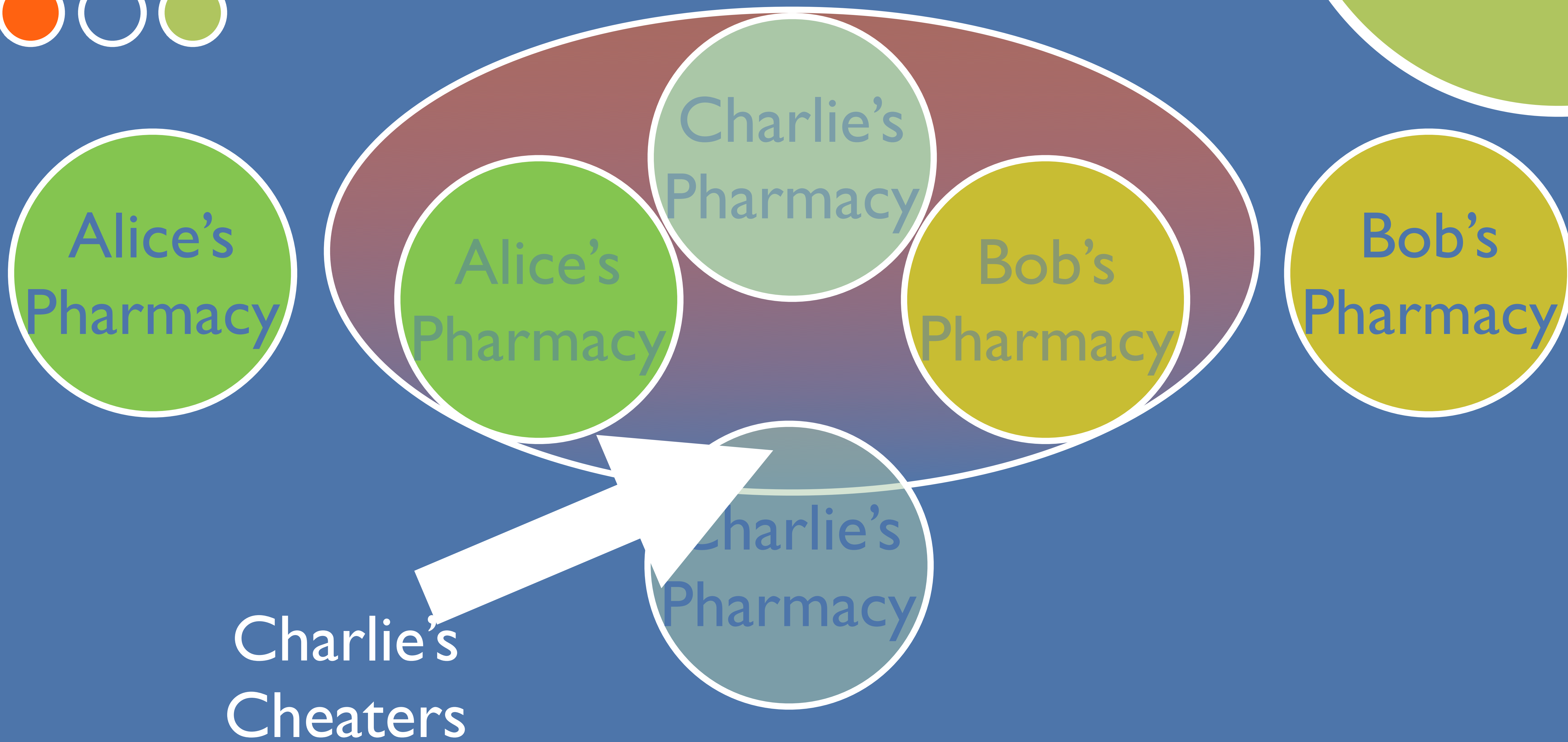
Prescription Cheaters



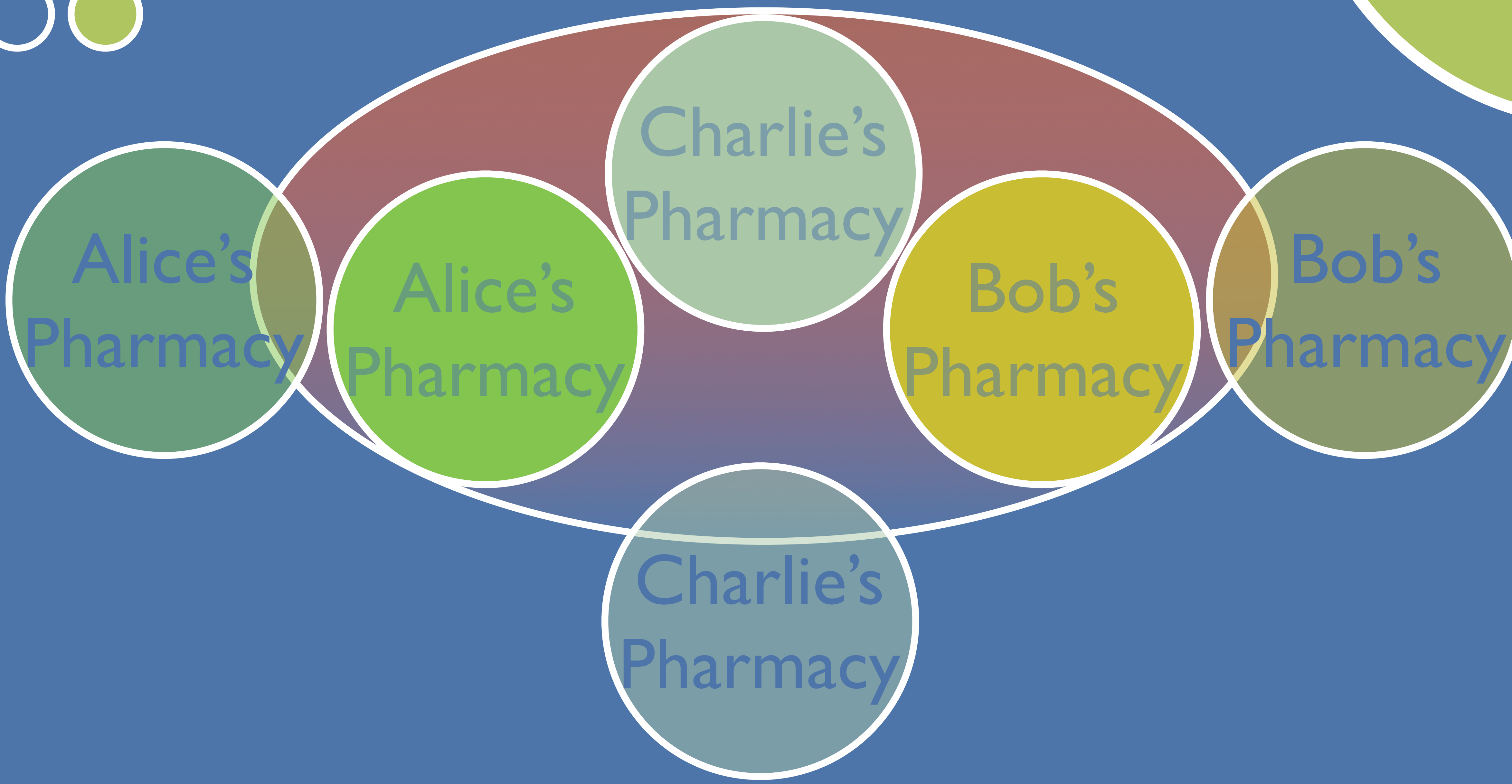
Prescription Cheaters



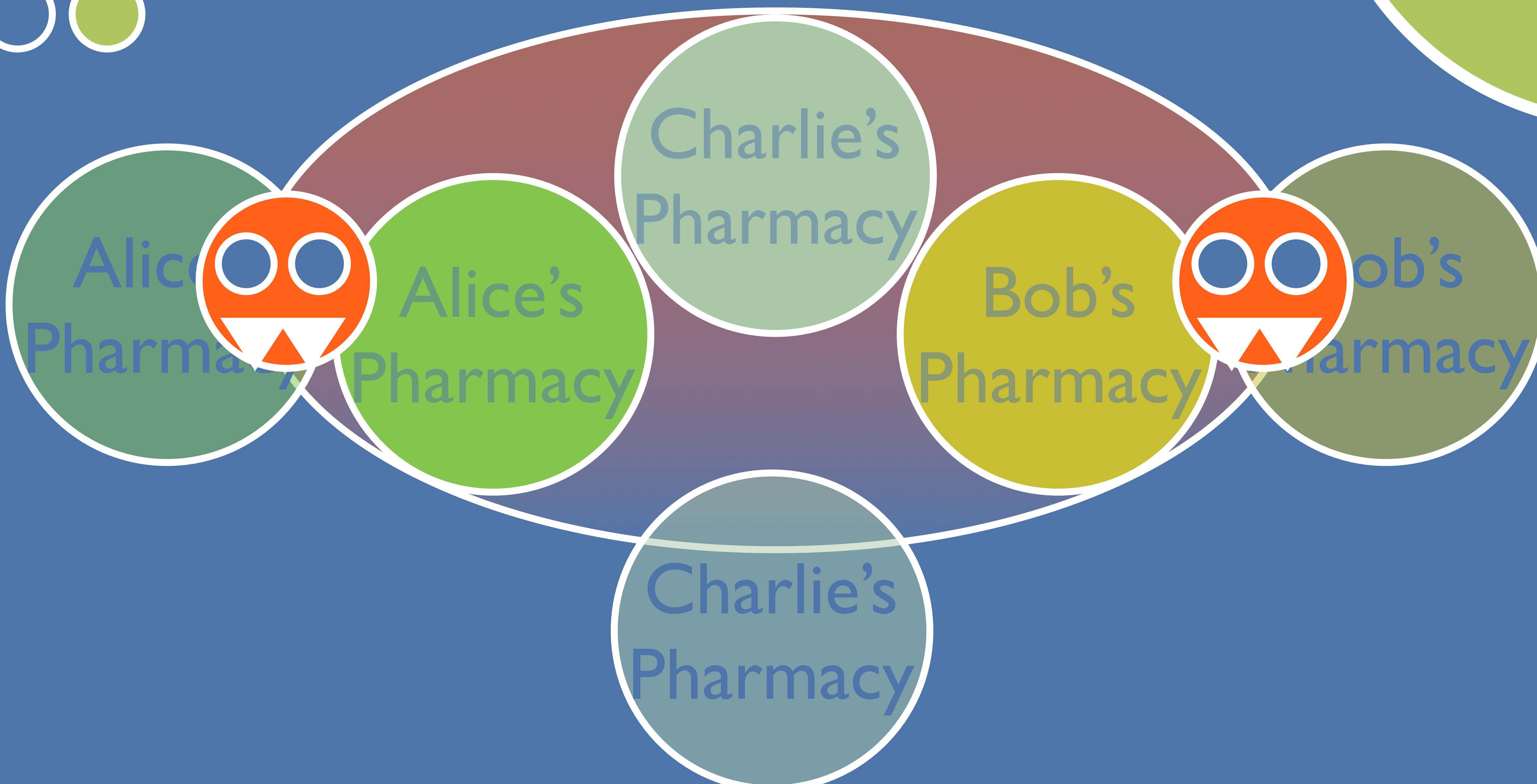
Prescription Cheaters



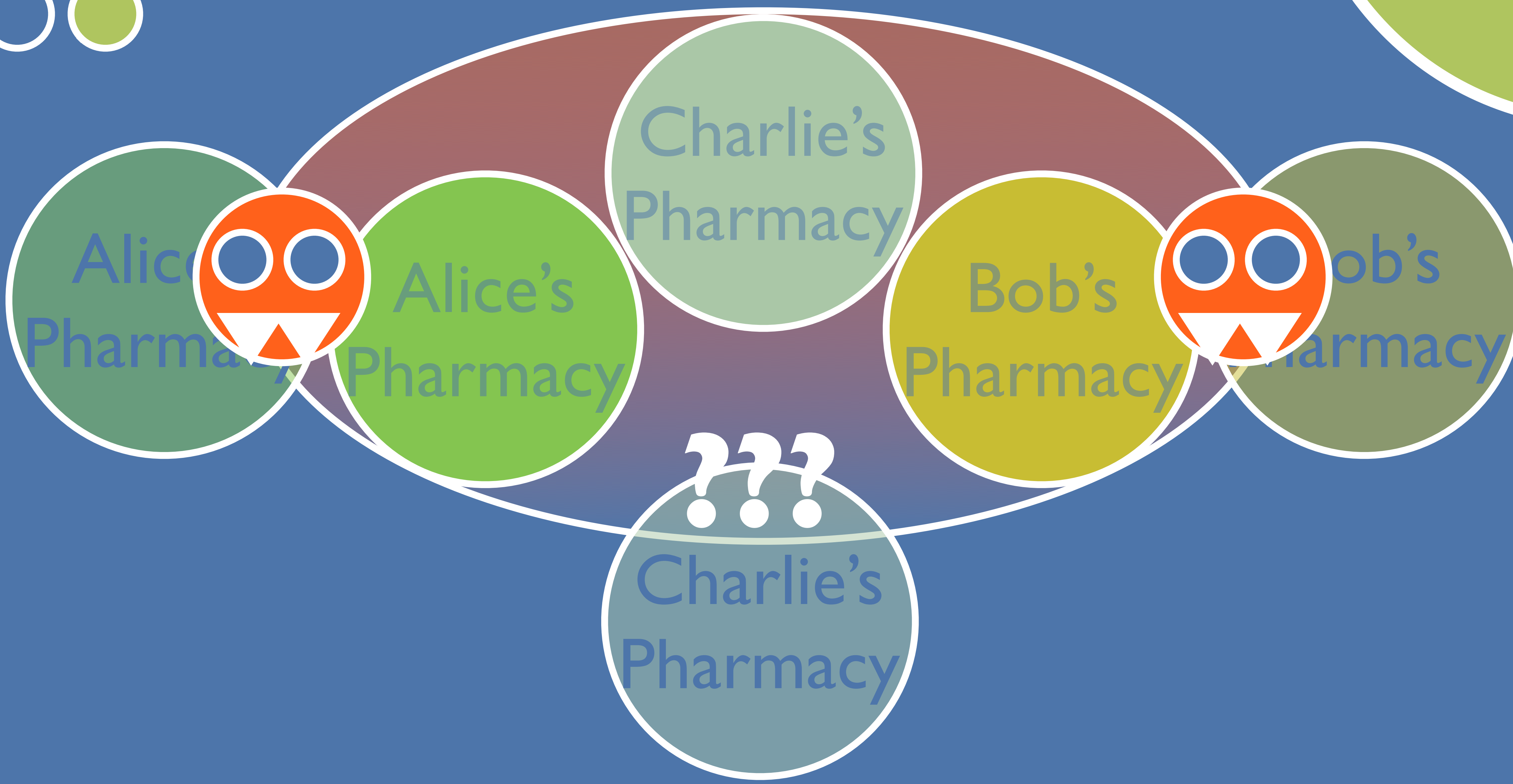
Prescription Cheaters



Prescription Cheaters

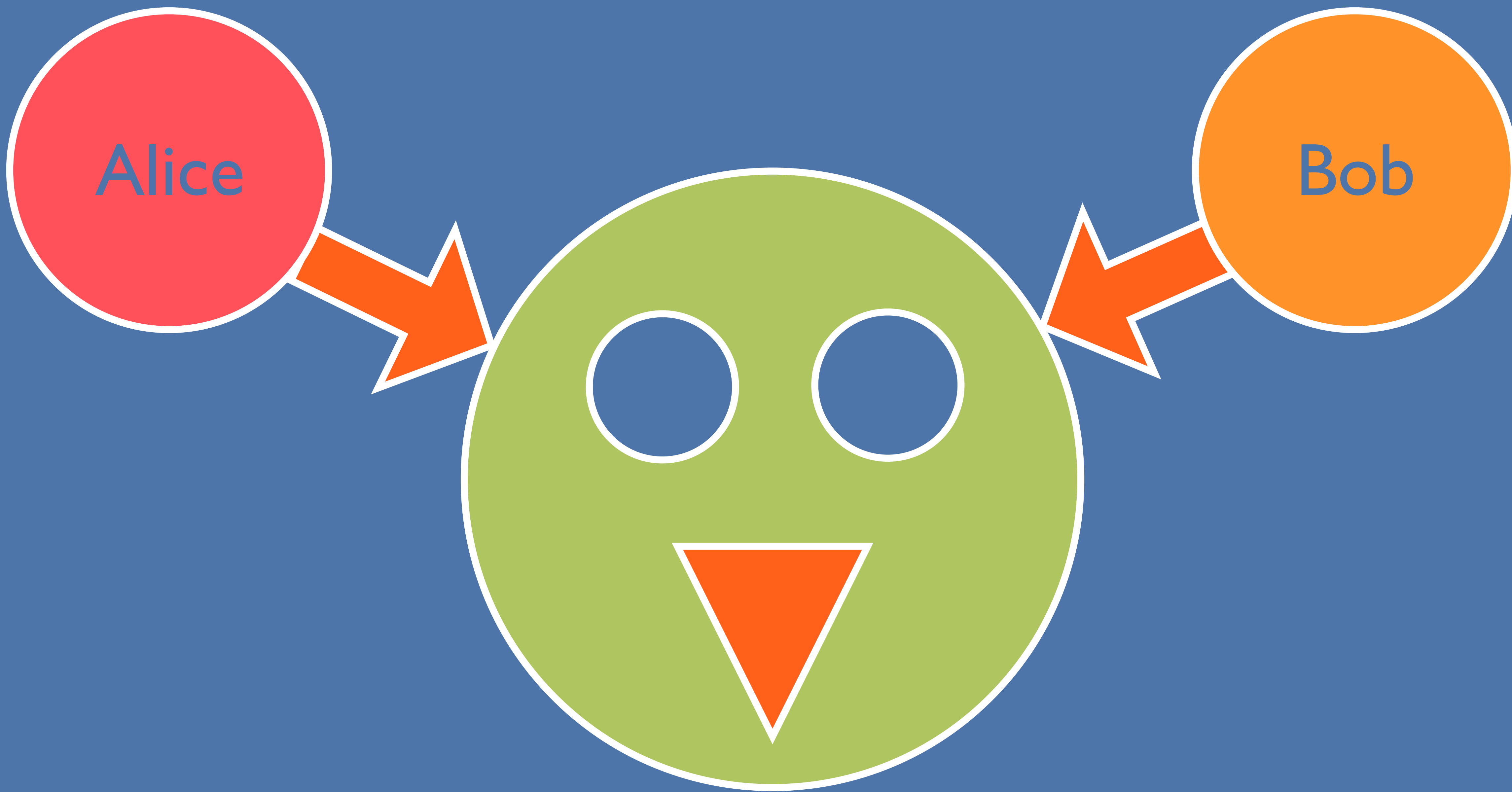


Prescription Cheaters



The Ideal Model

The Ideal Model



The Ideal Model

The Ideal Model

Perfectly secure
with trusted third party

The Ideal Model

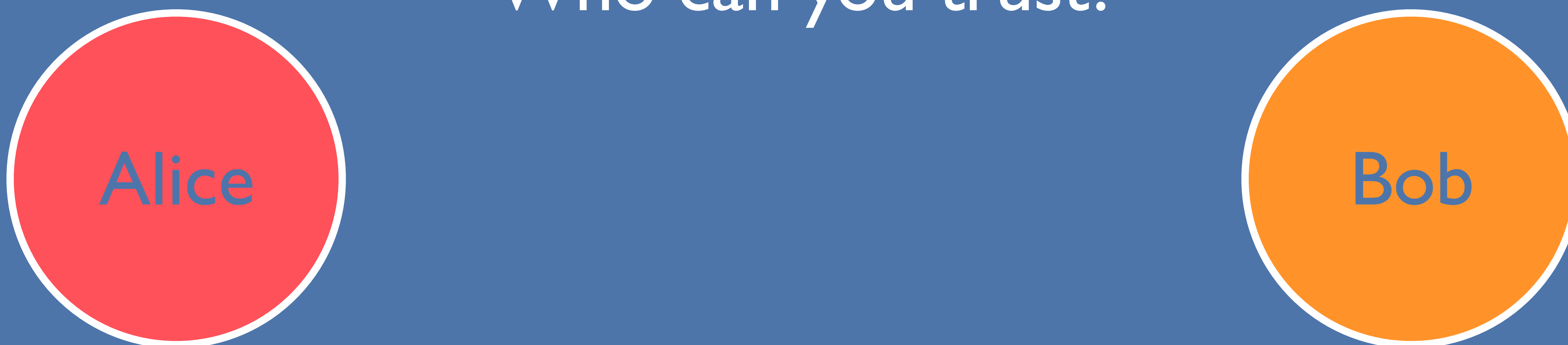
Who can you trust?

The Ideal Model

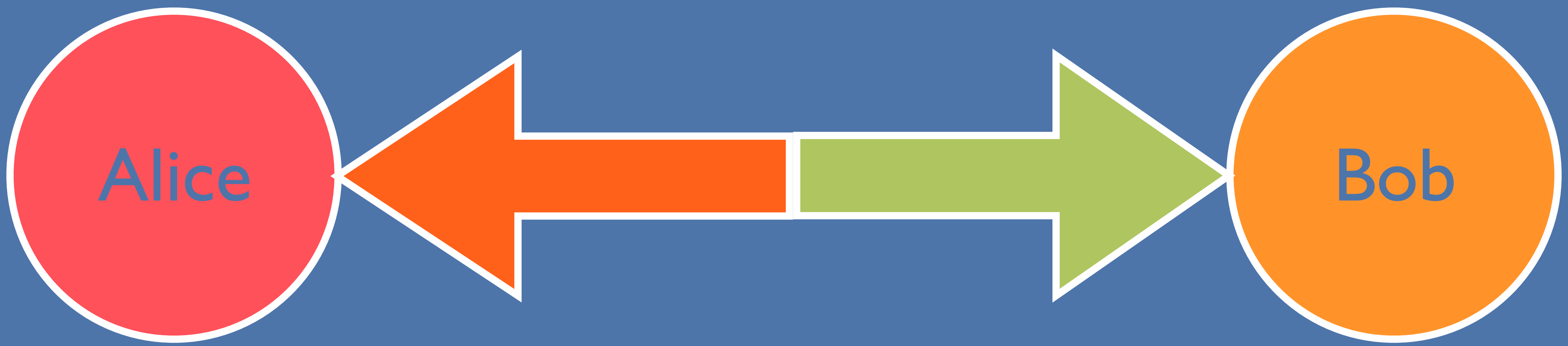
Who can you trust?

The Ideal Model

Who can you trust?



The Ideal Model



To increase real-world security, we remove the trusted party

Outline

- Motivational examples
- Multisets represented as polynomials
- Polynomial operations
- Multiset operations with polynomials
- Use of our techniques
- Contributions and related work

- We will represent all multisets as polynomials over a ring *R* (e.g. Z_{pq})
  - {a, b, c, c} → (x-a) (x-b) (x-c) (x-c)

- We will represent all multisets as polynomials over a ring *R* (e.g. Z_{pq})
  - {a, b, c, c} → (x-a) (x-b) (x-c) (x-c)

- We will represent all multisets as polynomials over a ring *R* (e.g. Z_{pq})
  - {a, b, c, c} → (x-a) (x-b) (x-c) (x-c)

- We will represent all multisets as polynomials over a ring R (e.g. Z_{pq})
  - $\{a, b, c, c\} \rightarrow (x-a)\ (x-b)\ (x-c)\ (x-c)$

- We will represent all multisets as polynomials over a ring *R* (e.g. Z_{pq})
  - $\{a, b, c, c\} \rightarrow (x-a)\ (x-b)\ (x-c)\ (x-c)$
- Random polynomial: each coefficient is distributed uniformly, independently in *R*
  - $r_0 + r_1 x + \ldots + r_n x^n$

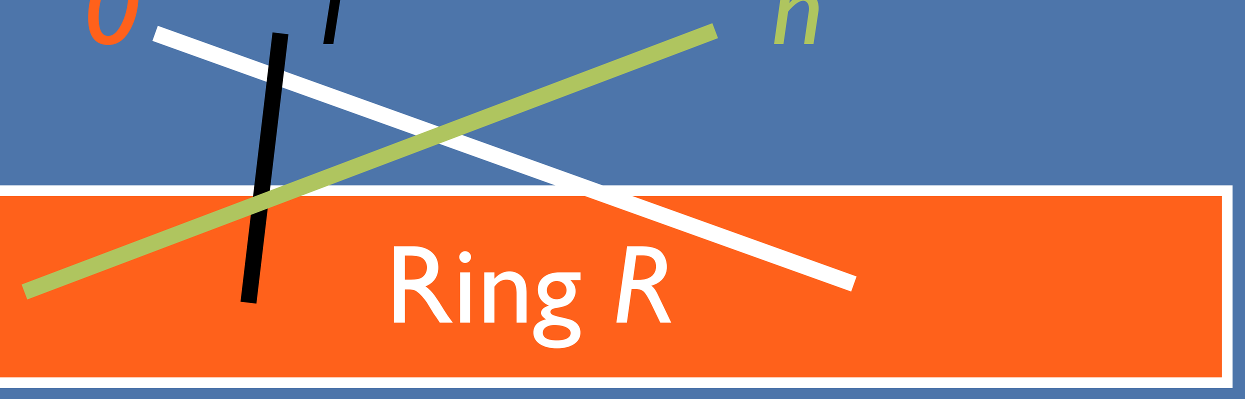
Ring *R*

- We will represent all multisets as polynomials over a ring R (e.g. Z_{pq})
  - $\{a, b, c, c\} \rightarrow (x-a)\ (x-b)\ (x-c)\ (x-c)$
- Random polynomial: each coefficient is distributed uniformly, independently in R
  - $r_0 + r_1 x + \ldots + r_n x^n$

Ring R

- We will represent all multisets as polynomials over a ring *R* (e.g. Z_{pq})
  - {a, b, c, c} → (x-a) (x-b) (x-c) (x-c)
- Random polynomial: each coefficient is distributed uniformly, independently in *R*
  - $r_0 + r_1 x + \ldots + r_n x^n$

- We will represent all multisets as polynomials over a ring *R* (e.g. Z_{pq})
  - $\{a, b, c, c\} \rightarrow (x-a)\,(x-b)\,(x-c)\,(x-c)$
- Random polynomial: each coefficient is distributed uniformly, independently in *R*
  - $r_0 + r_1 x + \ldots + r_n x^n$



- Random polynomials have random roots
- How can we ensure that we can recognize `random' elements?
  - We mark a small part of R as `valid'

Ring *R*

- Random polynomials have random roots
- How can we ensure that we can recognize `random' elements?
  - We mark a small part of R as `valid'

- Random polynomials have random roots
- How can we ensure that we can recognize `random' elements?
  - We mark a small part of R as `valid'
  - Thus random elements `look random' with overwhelming probability
  - One scheme: valid format *a||h(a)*

Polynomial Multiplication

- What happens when we multiply two polynomials?

(x-a)(x-b)(x-b)

*

(x-b)(x-c)

Polynomial Multiplication

- What happens when we multiply two polynomials?

(x-a)(x-b)(x-b)

*

(x-b)(x-c)

↓

(x-a)
(x-b)(x-b)(x-b)
(x-c)

Polynomial Multiplication

- What happens when we multiply two polynomials?
- The roots of *both* polynomials are preserved
- Multiplicity of roots is *additive*

$(x-a)(x-b)(x-b)$

$*$

$(x-b)(x-c)$

$(x-a)$
$(x-b)(x-b)(x-b)$
$(x-c)$

Polynomial Multiplication

- What happens when we multiply two polynomials?
- The roots of *both* polynomials are preserved
- Multiplicity of roots is *additive*

$$(x-a)(x-b)(x-b)$$

$$*$$

$$(x-b)(x-c)$$

$$\Downarrow$$

$$(x-a)$$
$$(x-b)(x-b)(x-b)$$
$$(x-c)$$

Polynomial Multiplication

- What happens when we multiply two polynomials?
- The roots of *both* polynomials are preserved
- Multiplicity of roots is *additive*

$(x-a)(x-b)(x-b)$

$*$

$(x-b)(x-c)$

$(x-a)$
$(x-b)(x-b)(x-b)$
$(x-c)$

Polynomial Multiplication

- What happens when we multiply two polynomials?
- The roots of *both* polynomials are preserved
- Multiplicity of roots is *additive*

$(x-a)(x-b)(x-b)$

$*$

$(x-b)(x-c)$

↓

$(x-a)$

$(x-b)(x-b)(x-b)$

$(x-c)$

Polynomial Multiplication

- What happens when we multiply two polynomials?
- The roots of *both* polynomials are preserved
- Multiplicity of roots is *additive*
- This operation acts like a union of multiset representations!

$$(x-a)(x-b)(x-b)$$
$$*$$
$$(x-b)(x-c)$$

$$\Downarrow$$

$$(x-a)$$
$$(x-b)(x-b)(x-b)$$
$$(x-c)$$

Polynomial Addition

- What happens when we add two polynomials?

$$(x-a)(x-b)(x-b)$$
$$+$$
$$(x-a)(x-b)(x-c)$$

Polynomial Addition

- What happens when we add two polynomials?

$$(x-a)(x-b)(x-b)$$
$$+$$
$$(x-a)(x-b)(x-c)$$

$$\Downarrow$$

$$(x-a)(x-b)*f$$

$$f(c)\neq 0$$

Polynomial Addition

- What happens when we add two polynomials?
- The *shared* roots of the polynomials are preserved
- The *minimum* multiplicity is preserved

$$(x-a)(x-b)(x-b)$$
$$+$$
$$(x-a)(x-b)(x-c)$$
$$\Downarrow$$
$$(x-a)(x-b)*f$$
$$f(c)\neq 0$$

Polynomial Addition

- What happens when we add two polynomials?
- The *shared* roots of the polynomials are preserved
- The *minimum* multiplicity is preserved

$$(x-a)(x-b)(x-b)$$
$$+$$
$$(x-a)(x-b)(x-c)$$

$$\Downarrow$$

$$(x-a)(x-b)*f$$

$$f(c)\neq 0$$

Polynomial Addition

- What happens when we add two polynomials?
- The *shared* roots of the polynomials are preserved
- The *minimum* multiplicity is preserved

$$(x-a)(x-b)(x-b)$$

$$+$$

$$(x-a)(x-b)(x-c)$$

$$\Downarrow$$

$$(x-a)(x-b)*f$$

$$f(c)\neq 0$$

Polynomial Addition

- What happens when we add two polynomials?
- The *shared* roots of the polynomials are preserved
- The *minimum* multiplicity is preserved

$$(x-a)(x-b)(x-b)$$
$$+$$
$$(x-a)(x-b)(x-c)$$

$$(x-a)(x-b)*f$$

$$f(c)\neq 0$$

Polynomial Addition

- What happens when we add two polynomials?
- The *shared* roots of the polynomials are preserved
- The *minimum* multiplicity is preserved
- This operations acts somewhat like a multiset intersection!

$$(x-a)(x-b)(x-b)$$
$$+$$
$$(x-a)(x-b)(x-c)$$

↓

$$(x-a)(x-b)*f$$

$$f(c)\neq 0$$

Polynomial Addition

- What happens when we add two polynomials?
- The *shared* roots of the polynomials are preserved
- The *minimum* multiplicity is preserved
- This operations acts somewhat like a multiset intersection!

$$(x-a)(x-b)(x-b)$$
$$+$$
$$(x-a)(x-b)(x-c)$$

gcd

$$(x-a)(x-b) * f$$

$$f(c) \neq 0$$

Polynomial Derivatives

- What happens when we take the derivative of a polynomial?

(x-a)
(x-b)(x-b)(x-b)
(x-c)(x-c)

Polynomial Derivatives

- What happens when we take the derivative of a polynomial?

$(x-a)$
$(x-b)(x-b)(x-b)$
$(x-c)(x-c)$

↓

$(x-b)(x-b)(x-c)*f$

$f(a) \neq 0$

Polynomial Derivatives

- What happens when we take the derivative of a polynomial?
- The multiplicity of each root is *reduced* by one

$(x-a)$
$(x-b)(x-b)(x-b)$
$(x-c)(x-c)$

↓

$(x-b)(x-b)(x-c) * f$

$f(a) \neq 0$

Polynomial Derivatives

- What happens when we take the derivative of a polynomial?
- The multiplicity of each root is *reduced* by one

$$(x-a)$$
$$(x-b)(x-b)(x-b)$$
$$(x-c)(x-c)$$

$$(x-b)(x-b)(x-c)*f$$

$$f(a)\neq 0$$

Polynomial Derivatives

- What happens when we take the derivative of a polynomial?
- The multiplicity of each root is *reduced* by one

$(x-a)$
$(x-b)(x-b)(x-b)$
$(x-c)(x-c)$

$(x-b)(x-b)(x-c)*f$

$f(a) \neq 0$

Polynomial Derivatives

- What happens when we take the derivative of a polynomial?
- The multiplicity of each root is *reduced* by one

$(x-a)$
$(x-b)(x-b)(x-b)$
$(x-c)(x-c)$

$(x-b)(x-b)(x-c)*f$

$f(a)\neq 0$

Polynomial Derivatives

- What happens when we take the derivative of a polynomial?
- The multiplicity of each root is *reduced* by one
- This acts somewhat like an *element reduction* operator!
- Note that I am glossing over some of the math...

$$(x-a)$$
$$(x-b)(x-b)(x-b)$$
$$(x-c)(x-c)$$

$$\downarrow$$

$$(x-b)(x-b)(x-c)*f$$

$$f(a)\neq 0$$

- We use these polynomial operations to calculate multiset union, intersection, and element reduction
- We cannot use the simple polynomial operations directly
  - They can reveal extra private information
    - e.g., elements that are not in the result set
  - The calculation can be manipulated by malicious players

- How can malicious players influence results?
  - If we are not careful about calculating intersection:

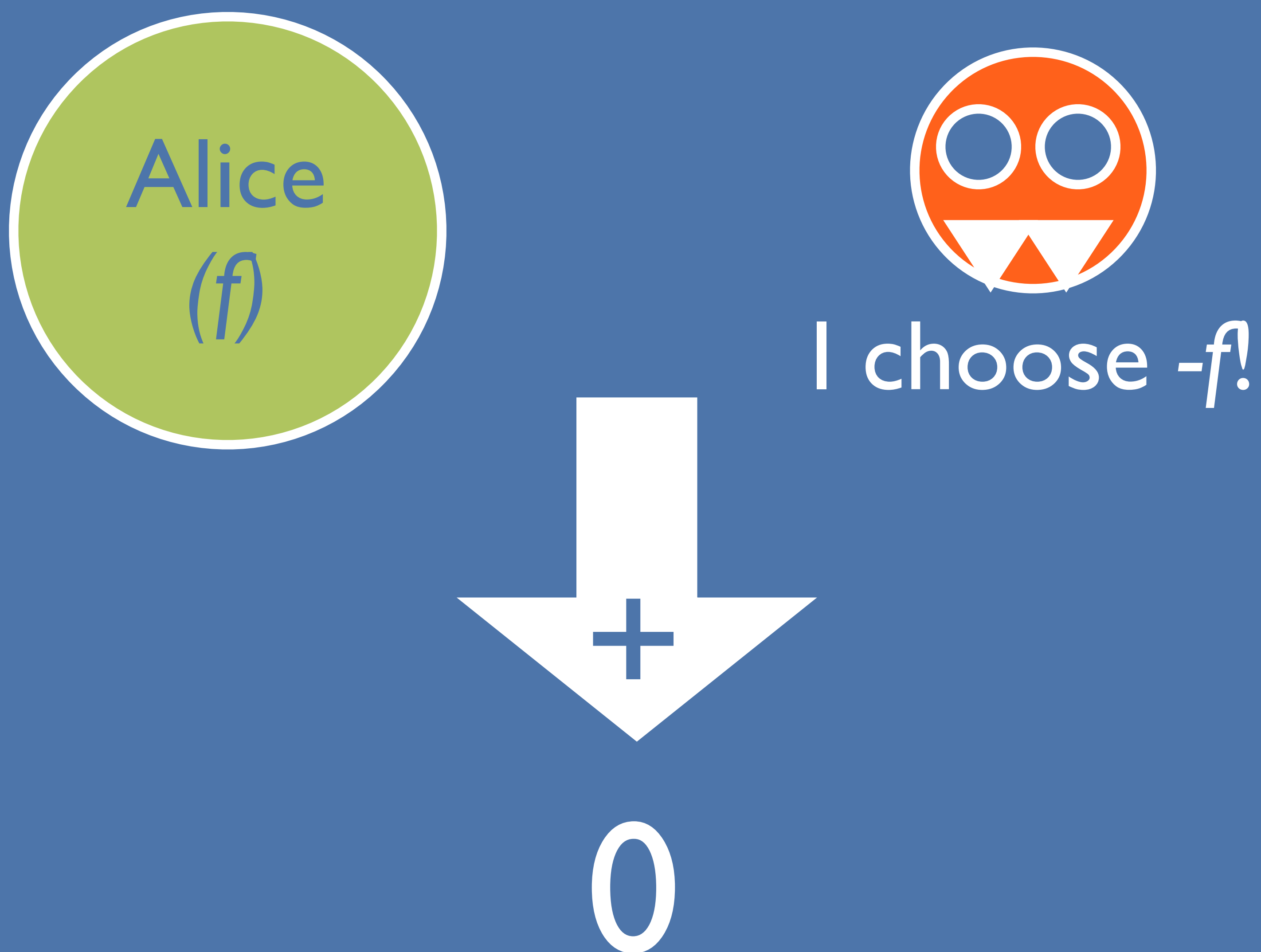
Alice
(*f*)

- How can malicious players influence results?
  - If we are not careful about calculating intersection:

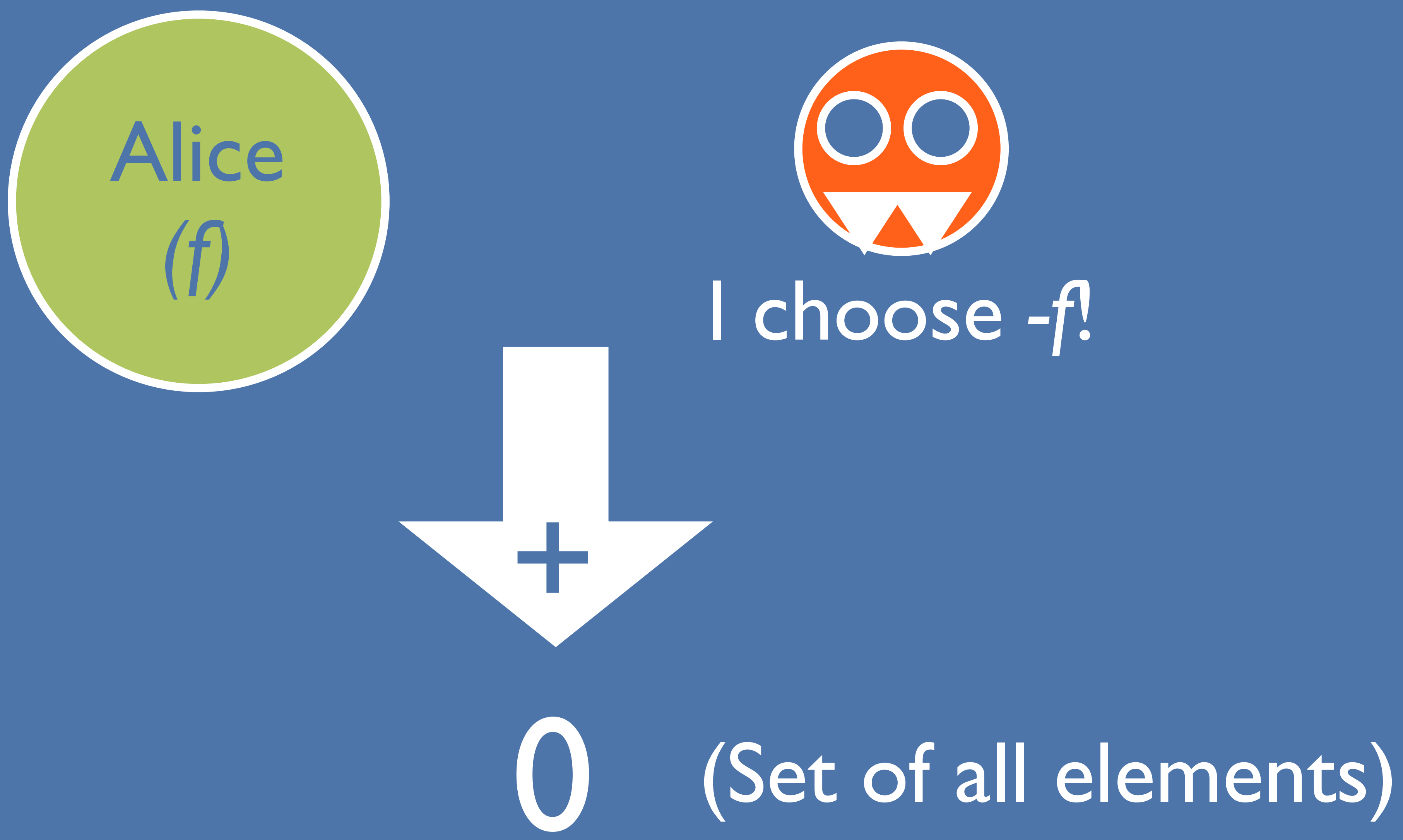
Alice
(*f*)

I choose -*f*!

- How can malicious players influence results?
  - If we are not careful about calculating intersection:

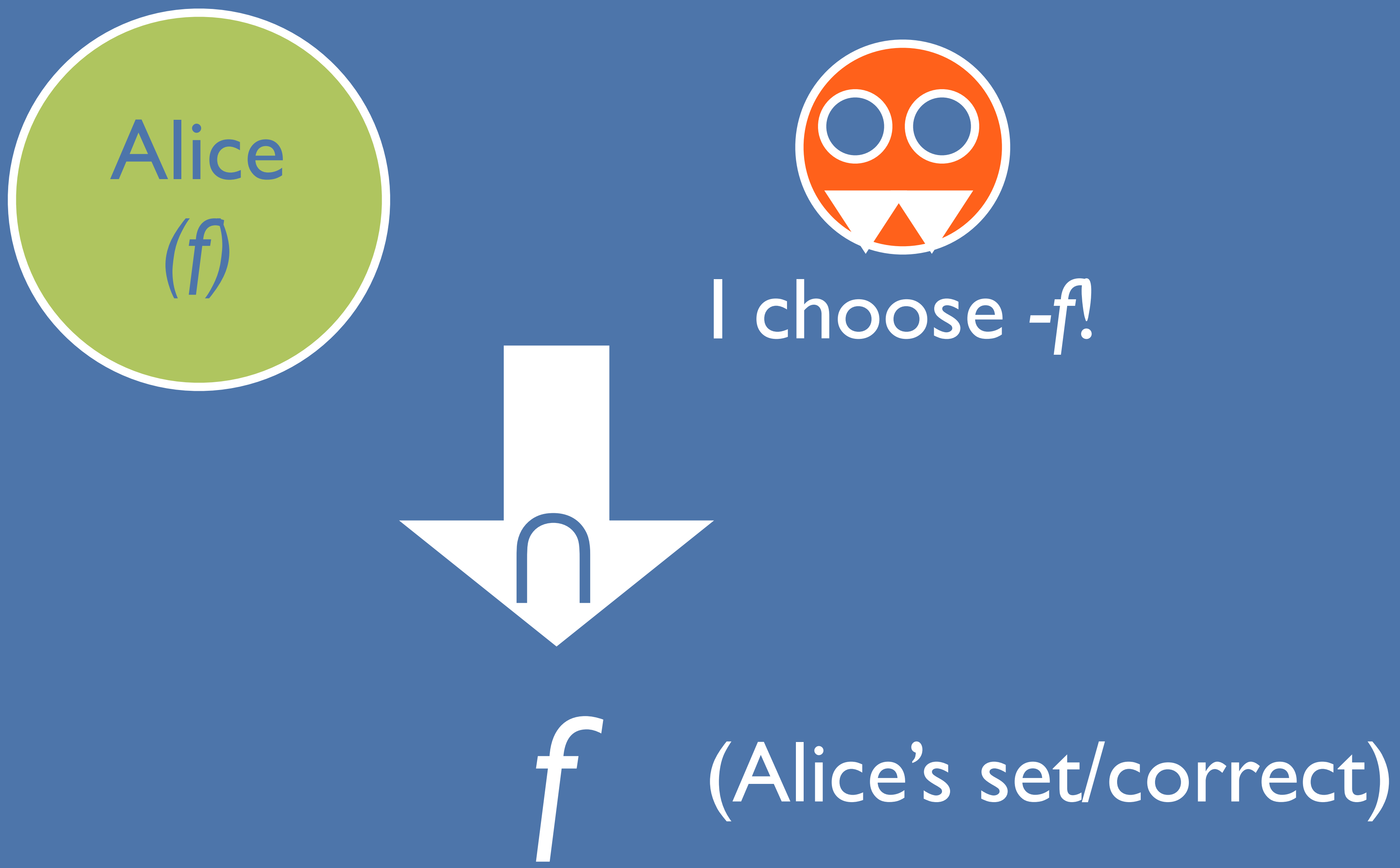


- How can malicious players influence results?
  - If we are not careful about calculating intersection:



- How can malicious players influence results?
  - If we are not careful about calculating intersection:

- How can malicious players influence results?
  - If we are not careful about calculating intersection:



- We must use randomness to hide `extra' information and enforce correctness

- We utilize the following lemma:
  - If **gcd(v,w)=1**, and **r,s are random** polynomials such that deg(v)=deg(w) ≤ size(r)=size(s)
  - Then **v∗r+w∗s is** a **random** polynomial

Union

$S \cup T$ is calculated as:

Let S, T be multisets represented by the polynomials f, g.

Union

SUT is calculated as:

f*g

Let S, T be multisets represented by the polynomials f, g.

Intersection

Let S, T be multisets represented by the polynomials f, g.
Let r, s be random polynomials.

S∩T is calculated as:

Intersection

Let S, T be multisets represented by the polynomials f, g.
Let r, s be random polynomials.

S∩T is calculated as:

$f*r+g*s$

Intersection

S∩T is calculated as:

$$f*r+g*s$$

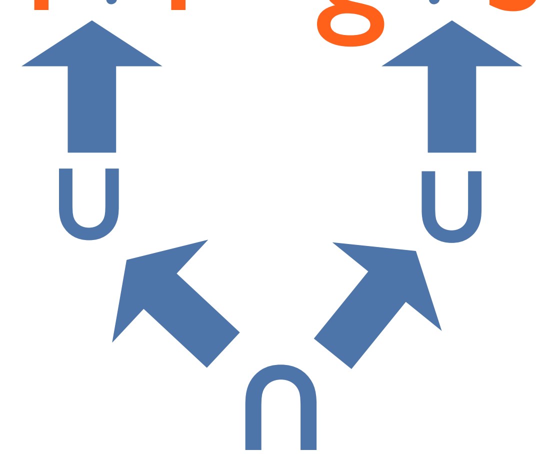
∪ ∪

Let S, T be multisets represented by the polynomials f, g.
Let r, s be random polynomials.

Intersection

S∩T is calculated as:

f∗r+g∗s



Let S, T be multisets represented by the polynomials f, g.
Let r, s be random polynomials.

Intersection

Let S, T be multisets represented by the polynomials f, g.
Let r, s be random polynomials.

S∩T is calculated as:

$f*r+g*s$

Intersection

Let S, T be multisets represented by the polynomials f, g.
Let r, s be random polynomials.

S∩T is calculated as:

$$f*r+g*s$$
$$=$$
$$\gcd(f,g)\,(w*r+v*s) =$$
$$\gcd(f,g)*u$$

Element Reduction

$Rd_t(S)$ is calculated as:

Let S be a multiset represented by the polynomial f.
Let r, s, F be random polynomials.

Element Reduction

$Rd_t(S)$ is calculated as:

$$f^{(t-1)} * F * r + f * s$$

Let S be a multiset represented by the polynomial f.
Let r, s, F be random polynomials.

Element Reduction

$Rd_t(S)$ is calculated as:

$$f^{(t-1)} * F * r + f * s$$

$$\cap$$

Let S be a multiset represented by the polynomial f.
Let r, s, F be random polynomials.

Element Reduction

$Rd_t(S)$ is calculated as:

$$f^{(t-1)} * F * r + f * s$$

Let S be a multiset represented by the polynomial f.
Let r, s, F be random polynomials.

Element Reduction

$Rd_t(S)$ is calculated as:

$$f^{(t-1)} * F * r + f * s =$$

$$\gcd(f^{(t-1)}, f)\,(w * r + v * s) =$$

$$\gcd(f^{(t-1)}, f) * u$$

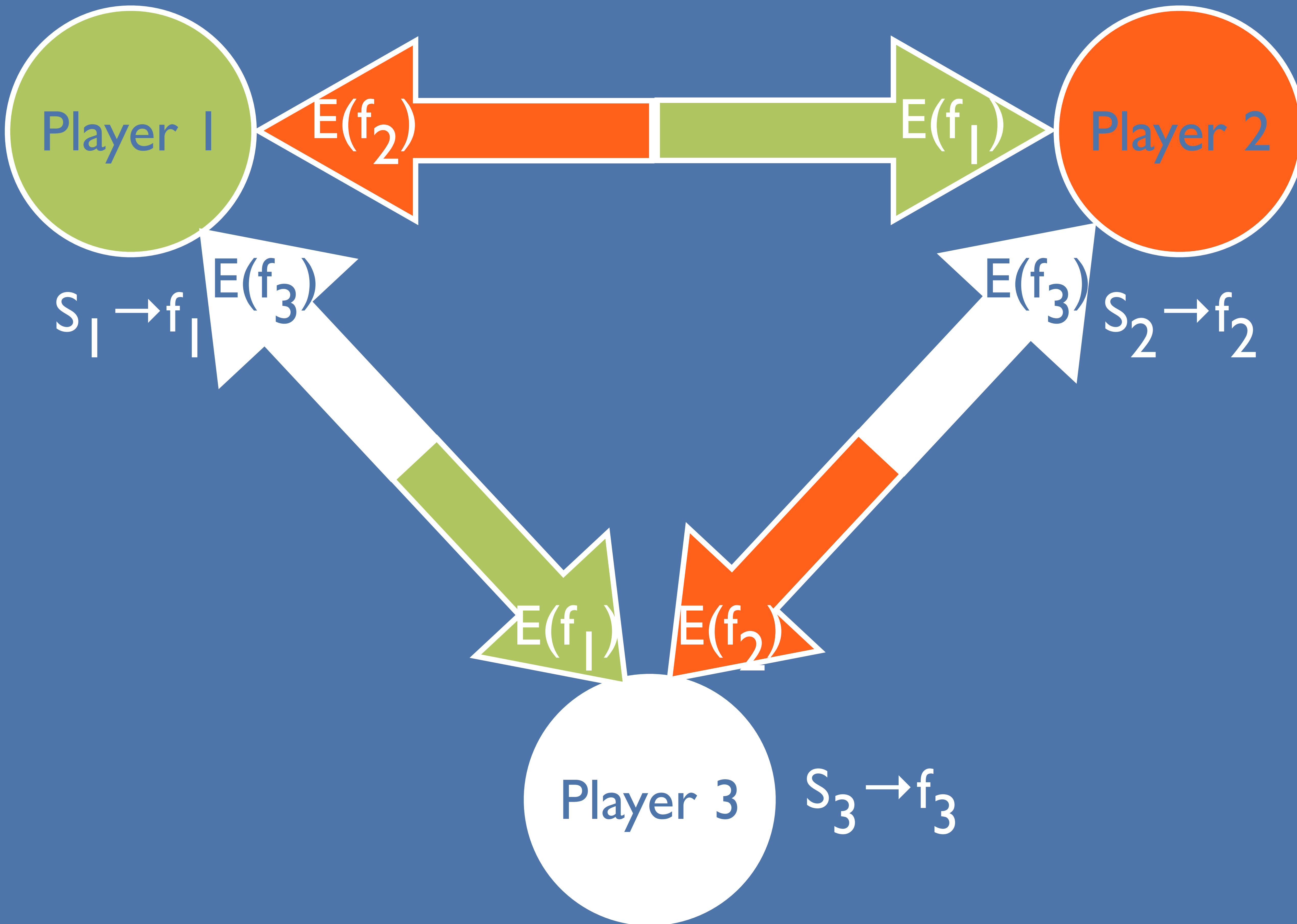
Let S be a multiset represented by the polynomial f.
Let r, s, F be random polynomials.

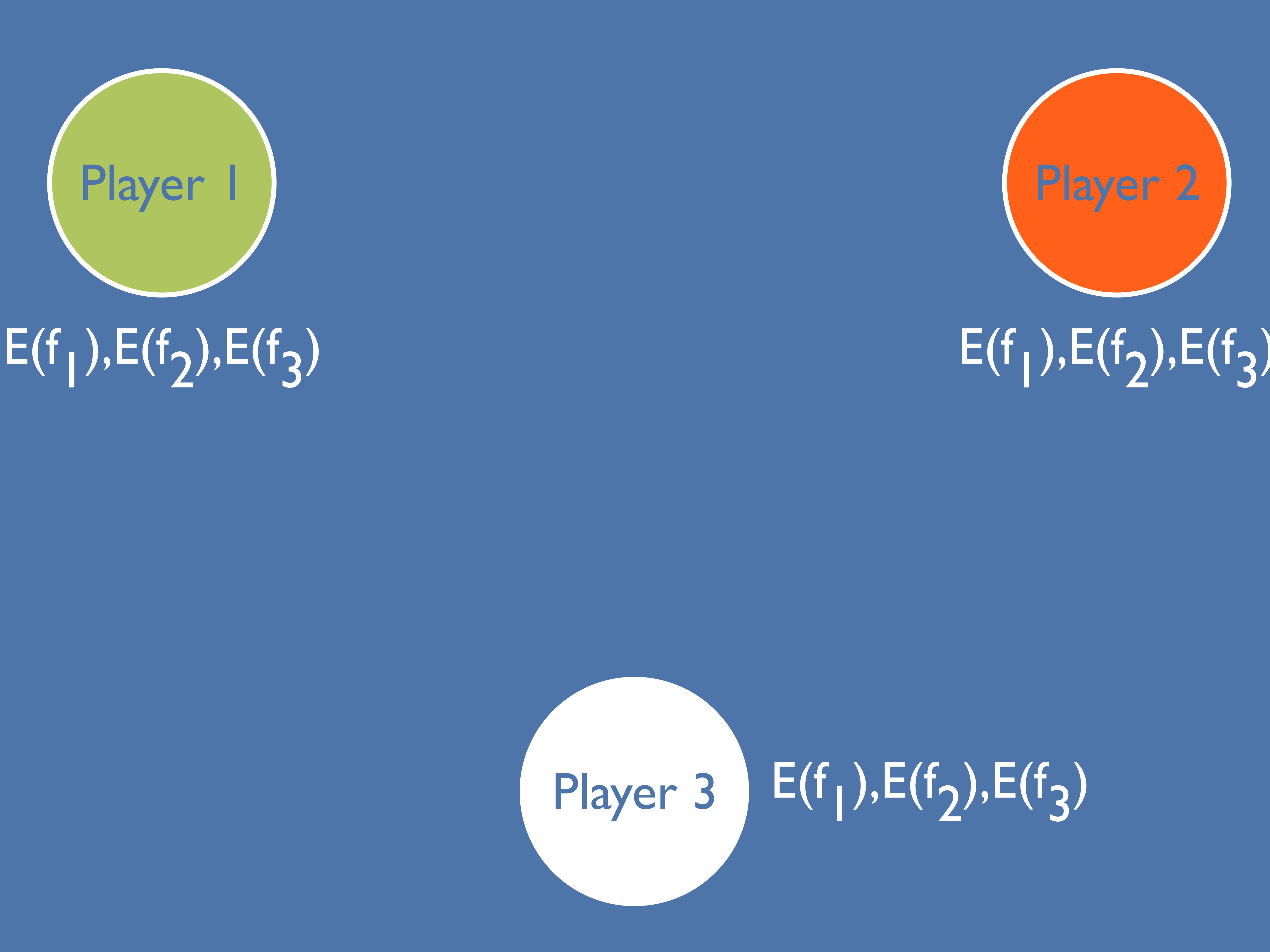
How do we use this?

- These techniques are not useful without the use of encryption
  - All players share a key
  - Special (homomorphic) cryptosystem
    - Addition, formal derivative of encrypted polynomials
    - Multiplication of known polynomial by encrypted polynomial


Player 1
S_1
Player 2
S_2
Multiset Intersection Protocol
Player 3
S_3


Player 1
$S_1 \rightarrow f_1$
Player 2
$S_2 \rightarrow f_2$
Player 3
$S_3 \rightarrow f_3$


Player 1
Player 2
Player 3
E(f2)
E(f1)
E(f3)
E(f3)
E(f1)
E(f2)
S1 → f1
S2 → f2
S3 → f3


Player 1
$E(f_1),E(f_2),E(f_3)$
Player 2
$E(f_1),E(f_2),E(f_3)$
Player 3
$E(f_1),E(f_2),E(f_3)$

Player 1

$E(f_1),E(f_2),E(f_3)$

Player 2

$E(f_1),E(f_2),E(f_3$

Each player i chooses random polynomials

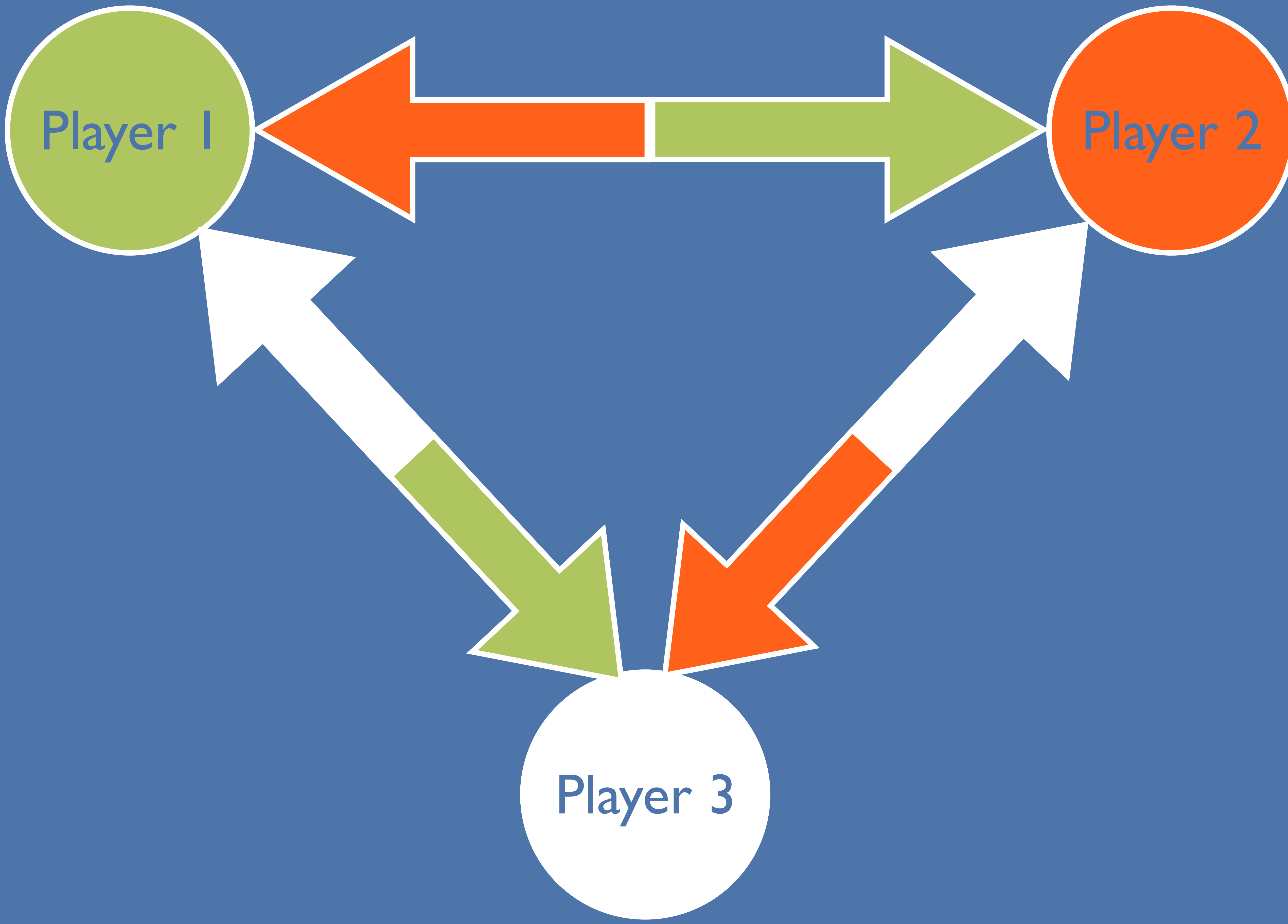
$r_{i,1}, r_{i,2}, r_{i,3}$

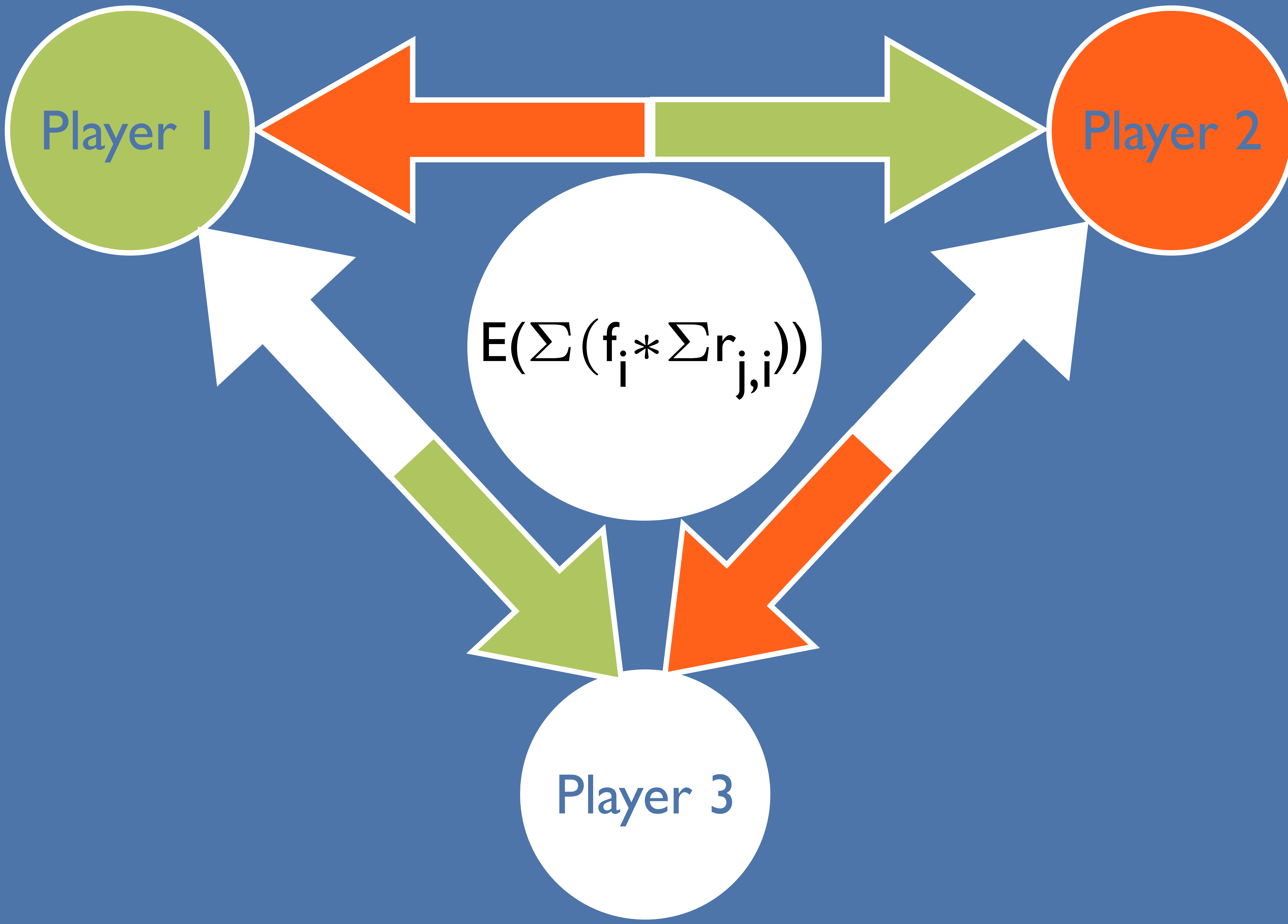
and calculates:

$E(f_1 * r_{i,1} + f_2 * r_{i,2} + f_3 * r_{i,3})$

Player 3

$E(f_1),E(f_2),E(f_3)$

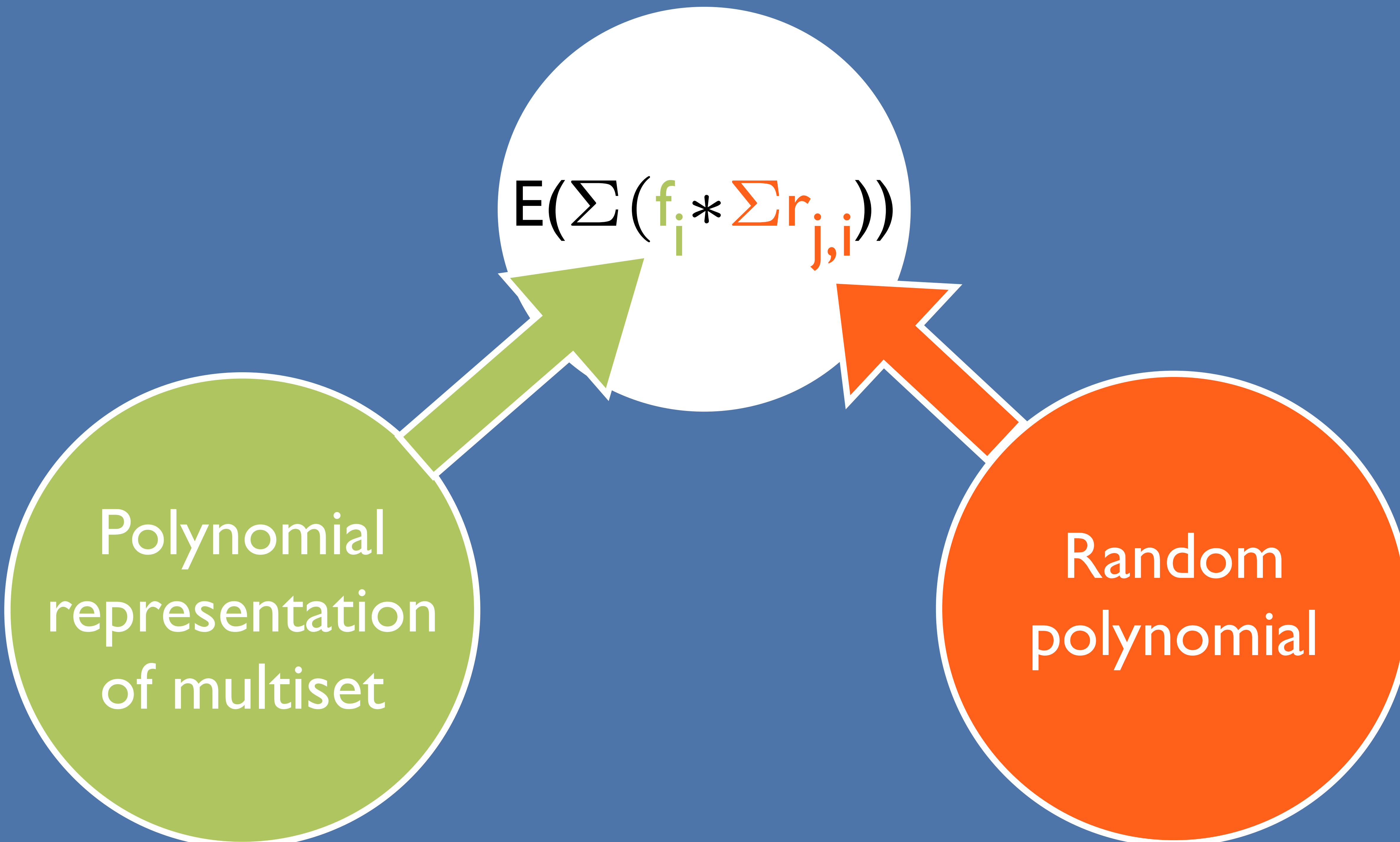

Player 1
Player 2
Player 3


Player 1
Player 2
$E(\Sigma(f_i * \Sigma r_{j,i}))$
Player 3

$$E(\sum(f_i * \sum r_{j,i}))$$

$$E(\Sigma(f_i * \Sigma r_{j,i}))$$

Polynomial representation of multiset


$E(\Sigma(f_i * \Sigma r_{j,i}))$
Polynomial representation of multiset
Random polynomial

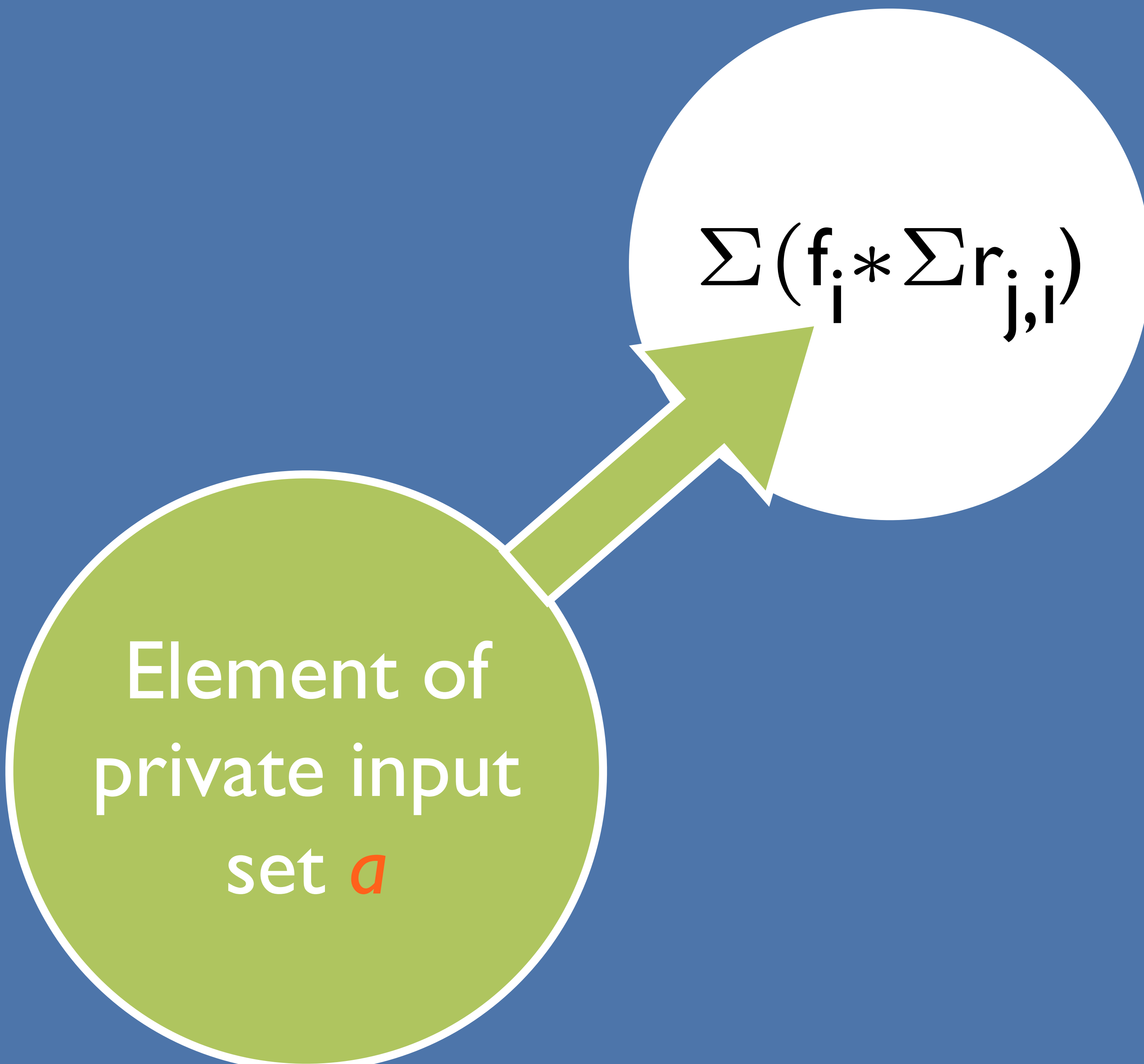
$$E(\Sigma(f_i * \Sigma r_{j,i}))$$

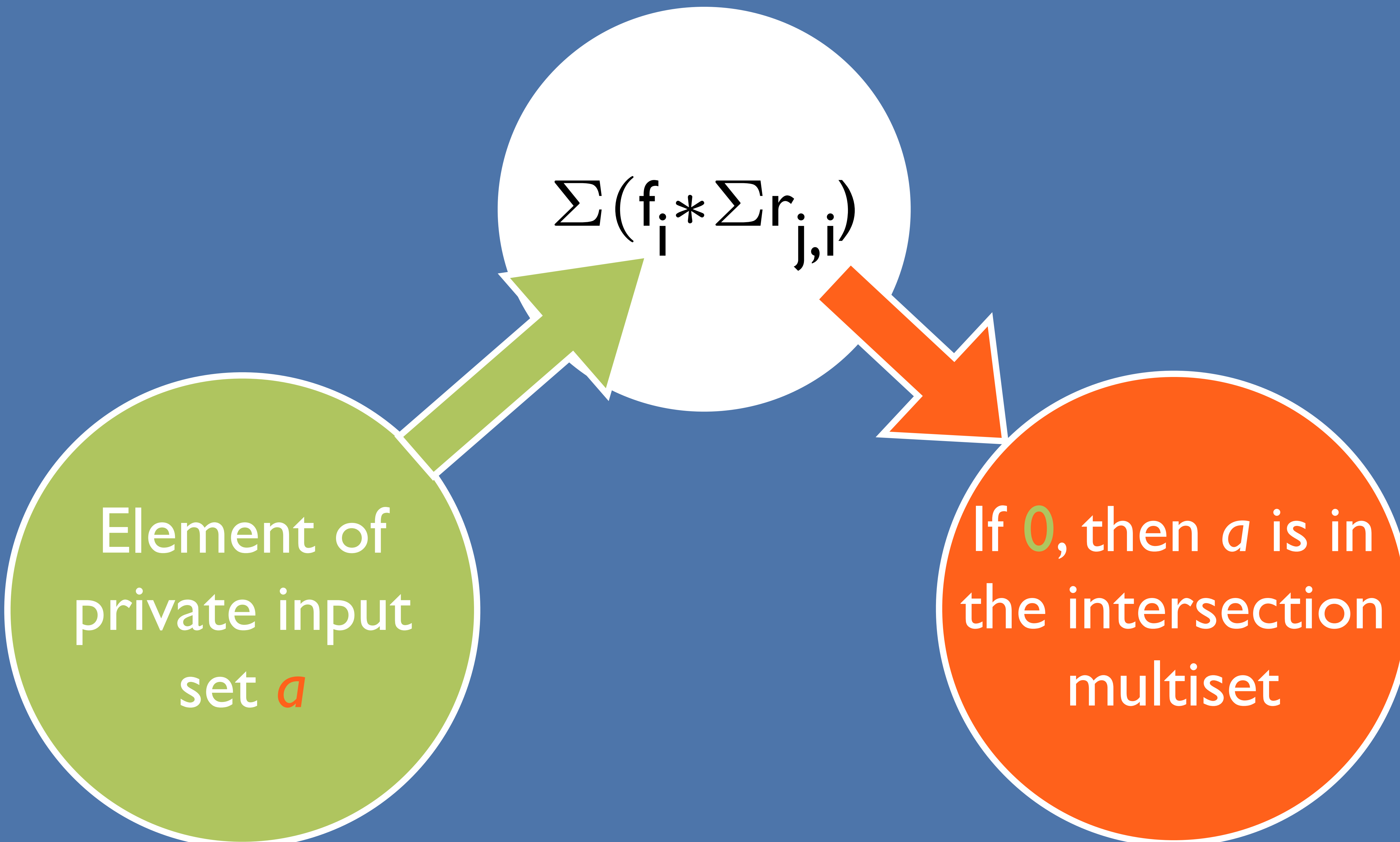
The players have calculated an encrypted polynomial representation of the multiset intersection!

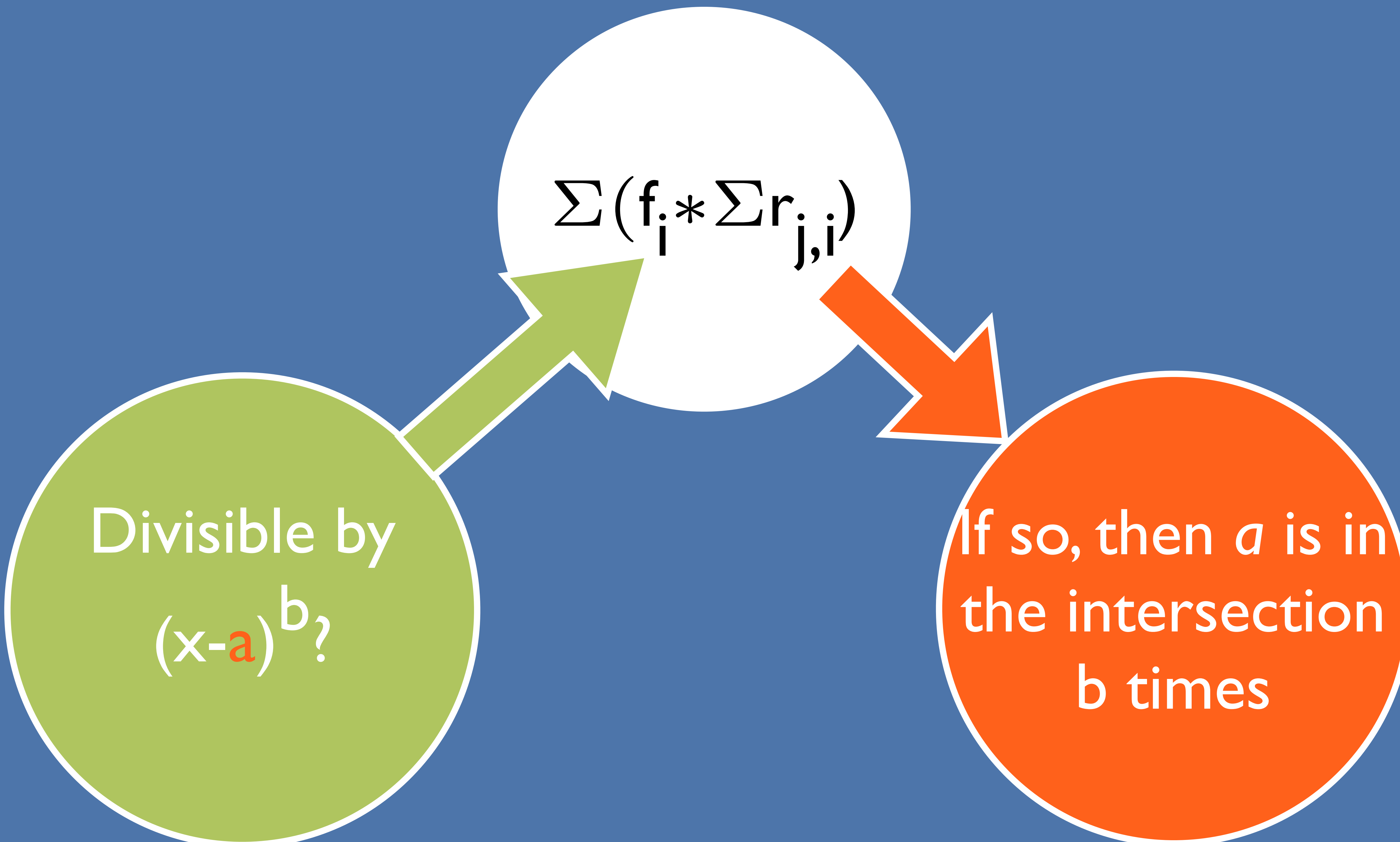
$$E(\Sigma(f_i * \Sigma r_{j,i}))$$

The players decrypt the polynomial, using their shared key.

$$\Sigma(f_i * \Sigma r_{j,i})$$


$\Sigma(f_i * \Sigma r_{j,i})$
Element of private input set *a*


$\Sigma(f_i * \Sigma r_{j,i})$
Element of private input set *a*
If 0, then *a* is in the intersection multiset


$\Sigma(f_i * \Sigma r_{j,i})$
Divisible by
$(x-a)^b$?
If so, then *a* is in
the intersection
b times

Outline

- Motivational examples
- Multisets represented as polynomials
- Polynomial operations
- Multiset operations with polynomials
- Use of our techniques
- Contributions and related work

- We have presented efficient, composable techniques for multiset intersection, union, and element reduction
- We design fair protocols for n≥2 players (malicious or HBC) for many set problems, including cardinality
- We design a protocol for determining subset relations
- We even evaluate boolean formulae!

- Two party set intersection (and related problems) [AES03] [FNP04]
- Set disjointness [KM05]
- Single-element-set intersection [FNW96] [NP99] [BST01] [L03]
- For most of the problems we address, the best previous result is through general MPC [Y82] [BGW88]

Thank you!