MEMBERSHIP DIMENSION

CATEGORY-BASED ROUTING IN SOCIAL NETWORKS
AND THE
SMALL-WORLD PHENOMENON

David Eppstein ◆ Michael Goodrich ◆ Maarten Löffler
Darren Strash ◆ Lowell Trott

PART I
INTRODUCTION

SMALL-WORLD PHENOMENON

SMALL-WORLD PHENOMENON

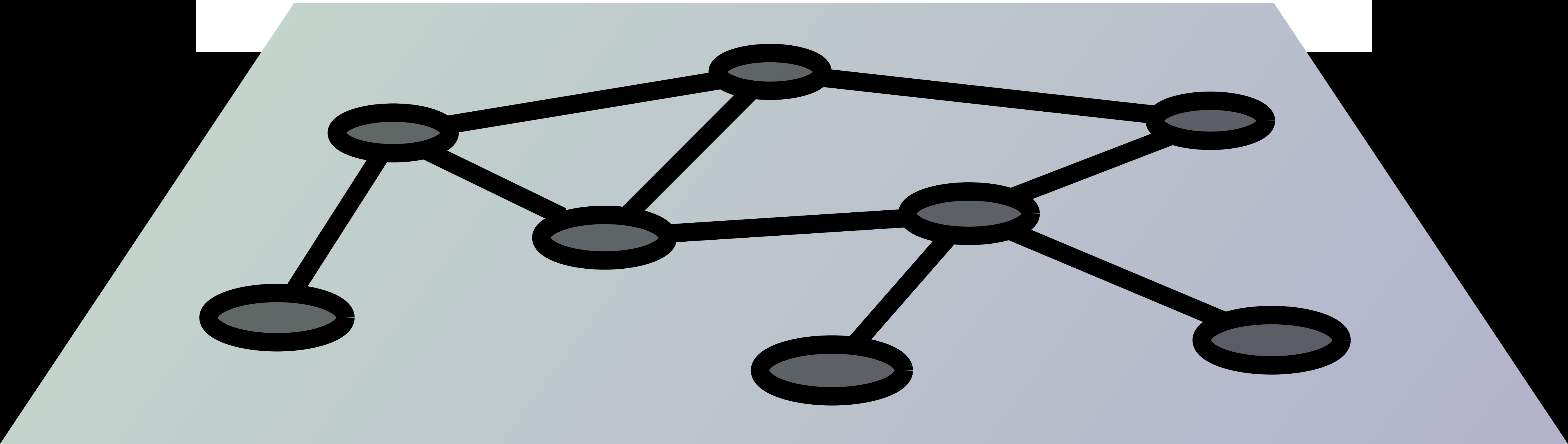
- Consider a *social network*

SMALL-WORLD PHENOMENON

- Consider a *social network*

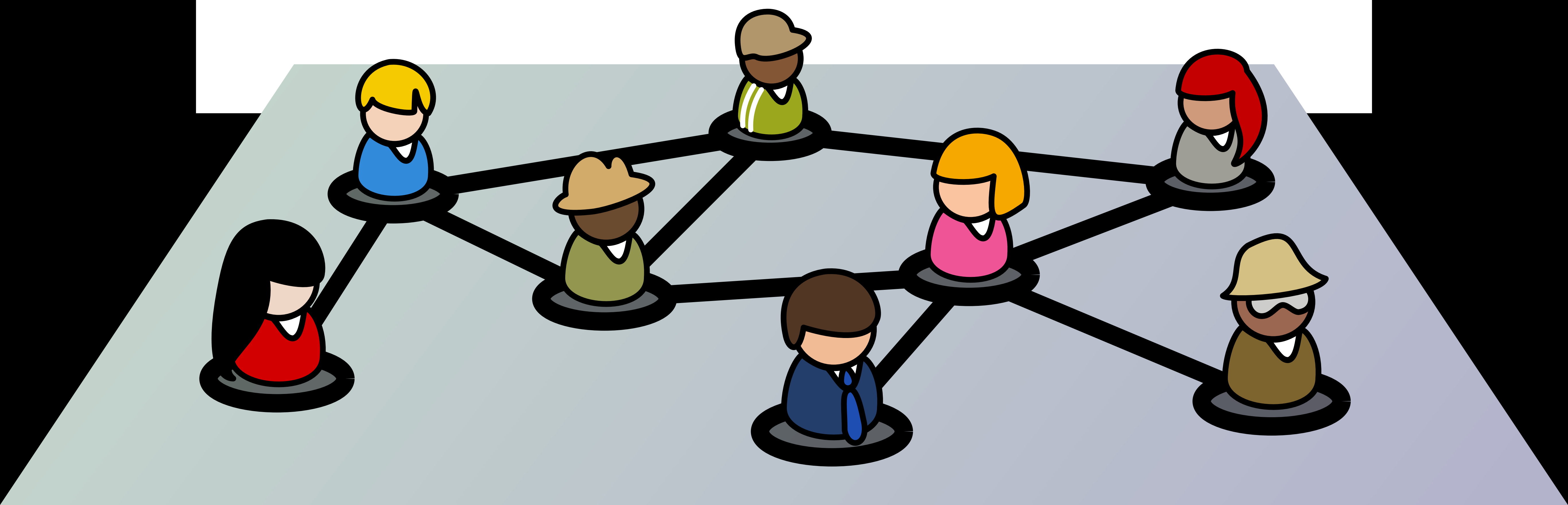
SMALL-WORLD PHENOMENON

- Consider a *social network*



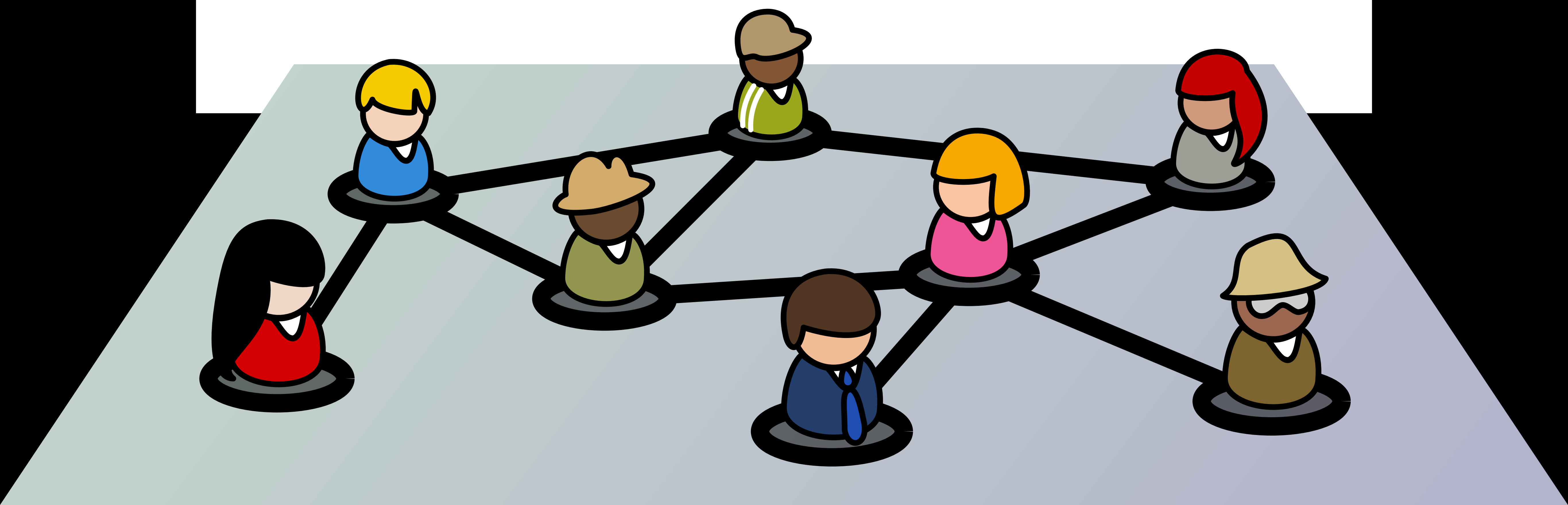
SMALL-WORLD PHENOMENON

- Consider a *social network*



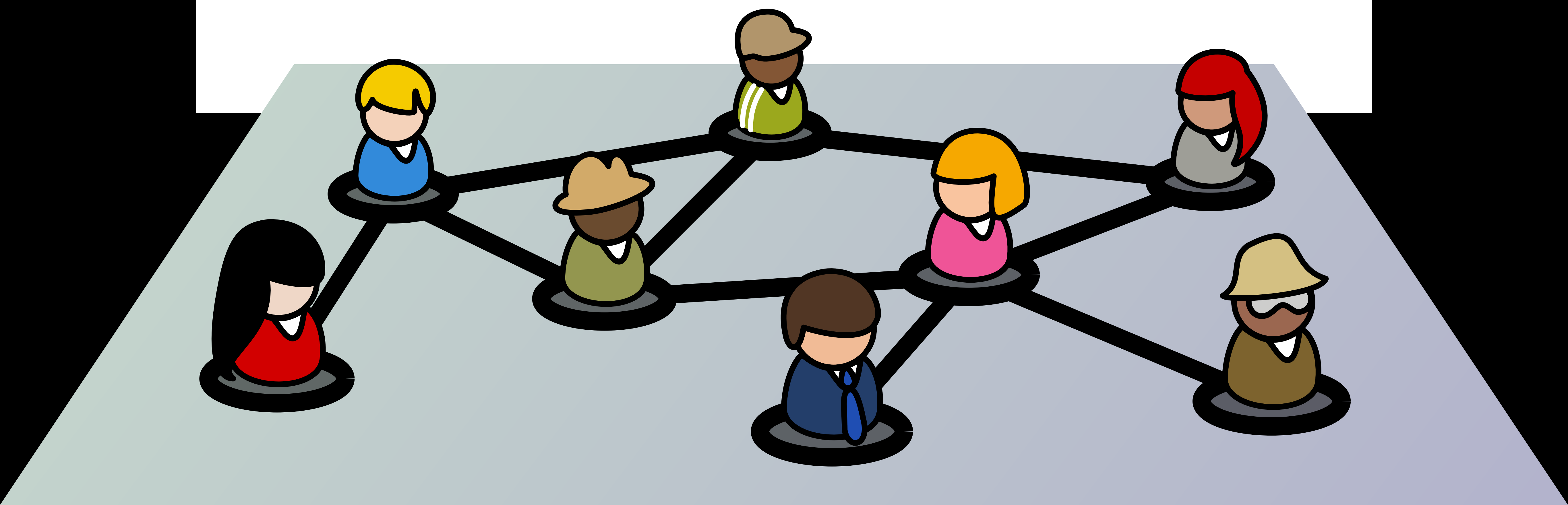
SMALL-WORLD PHENOMENON

- Consider a *social network*
- Milgram experiment [Milgram, 1967]



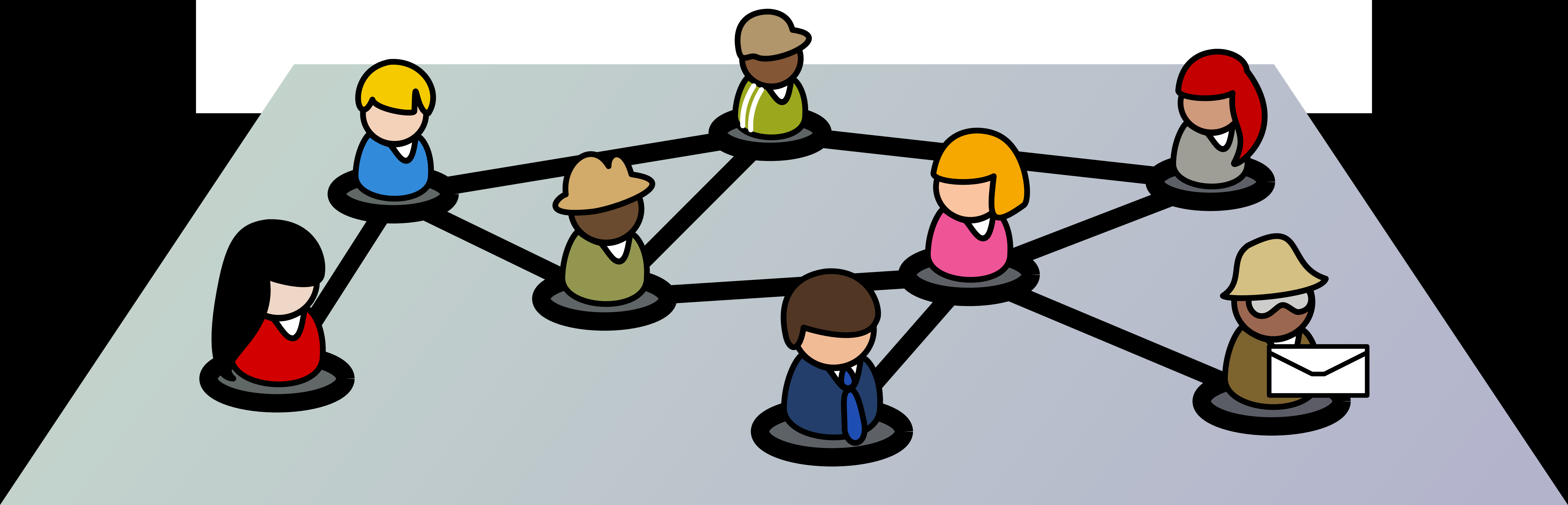
SMALL-WORLD PHENOMENON

- Consider a *social network*
- Milgram experiment [Milgram, 1967]
  - Give letter to random person



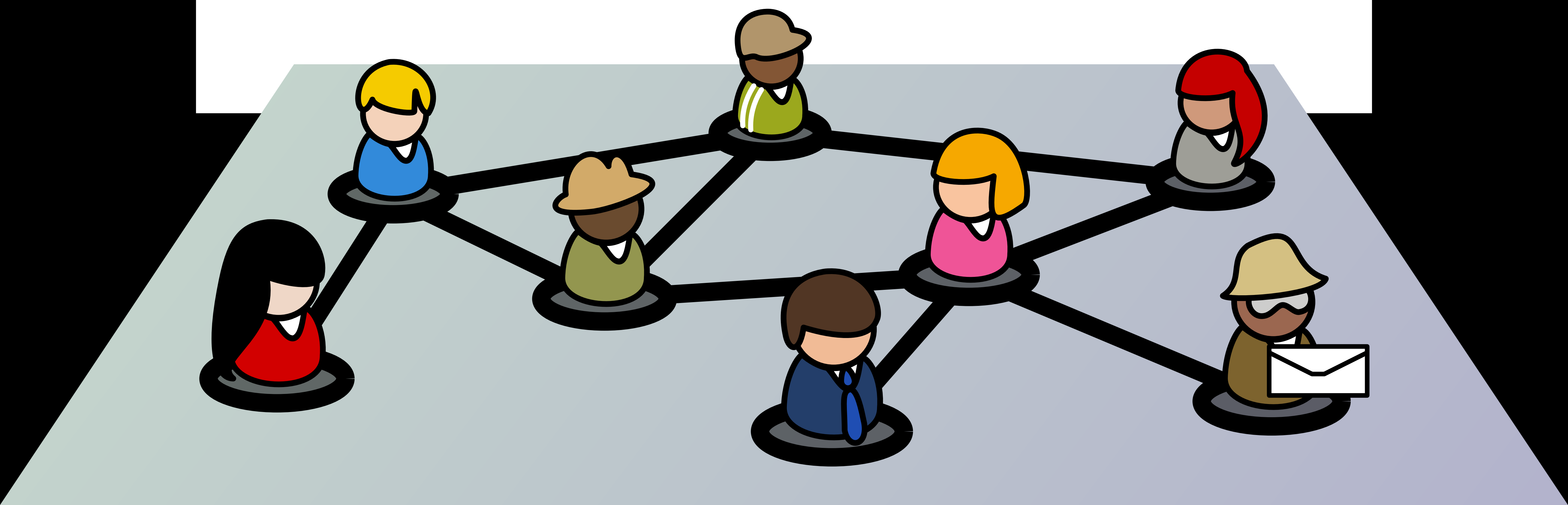
SMALL-WORLD PHENOMENON

- Consider a *social network*
- Milgram experiment [Milgram, 1967]
  - Give letter to random person



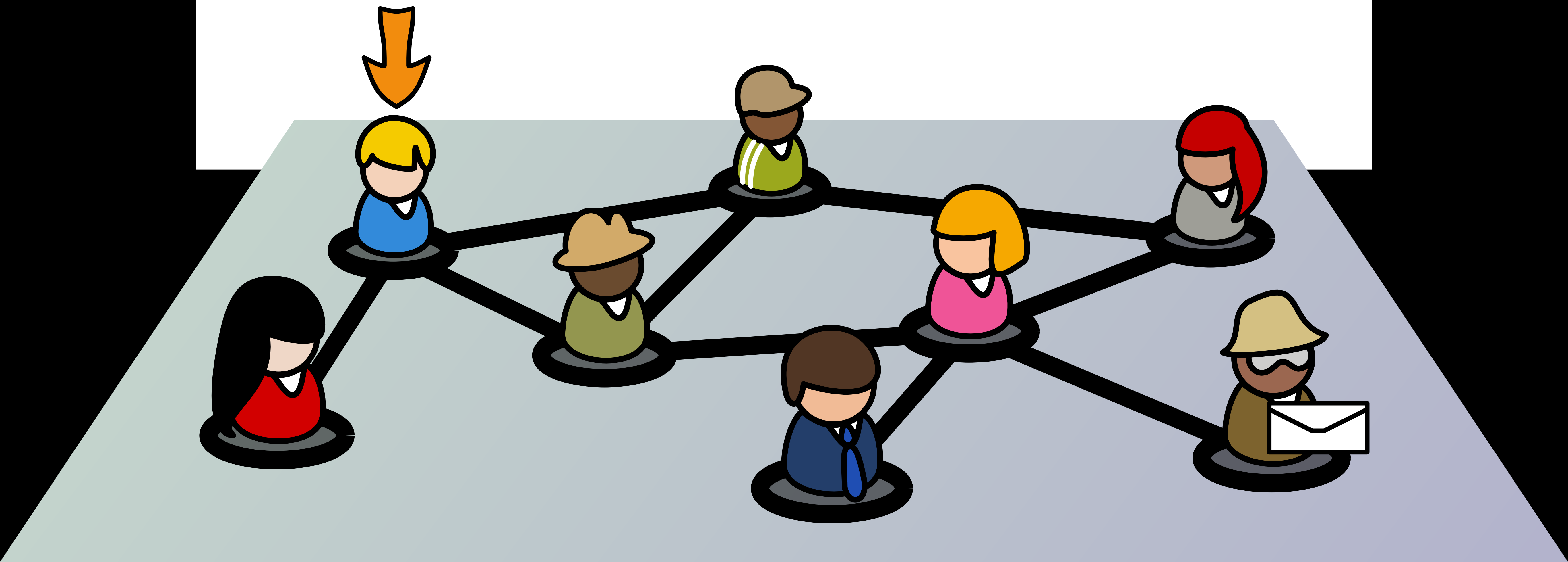
SMALL-WORLD PHENOMENON

- Consider a *social network*
- Milgram experiment [Milgram, 1967]
  - Give letter to random person
  - Select a random target



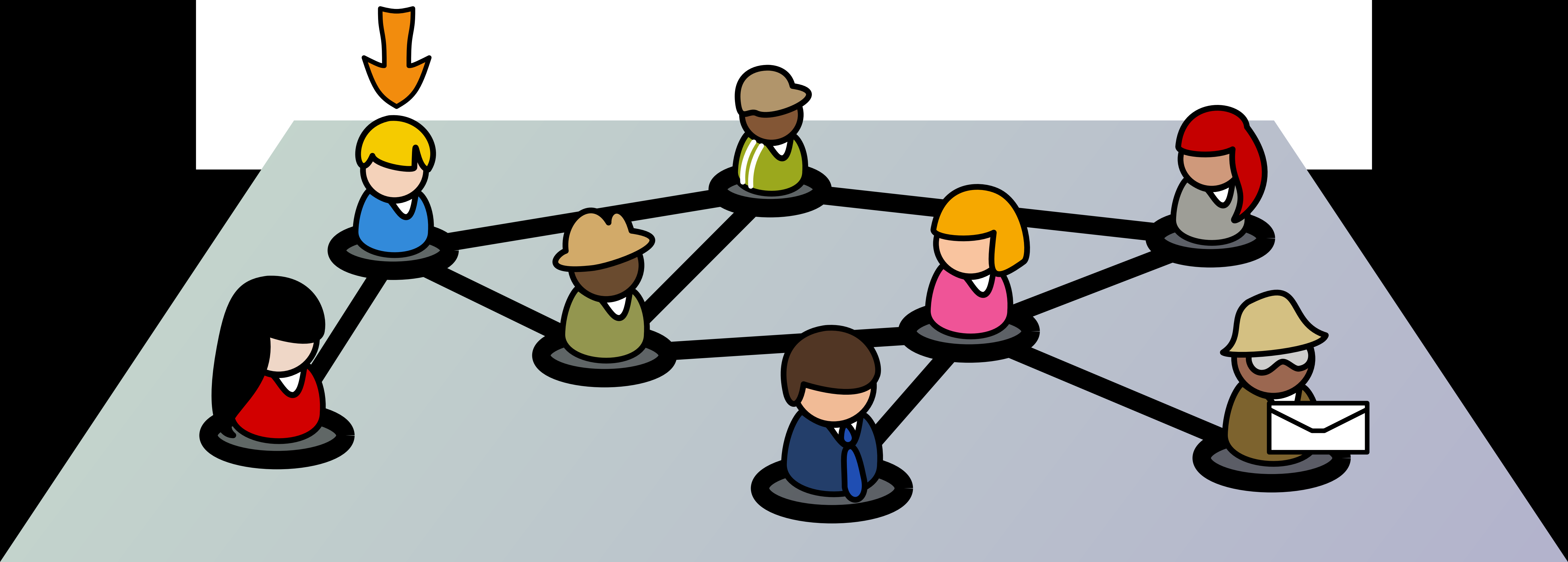
SMALL-WORLD PHENOMENON

- Consider a *social network*
- Milgram experiment [Milgram, 1967]
  - Give letter to random person
  - Select a random target



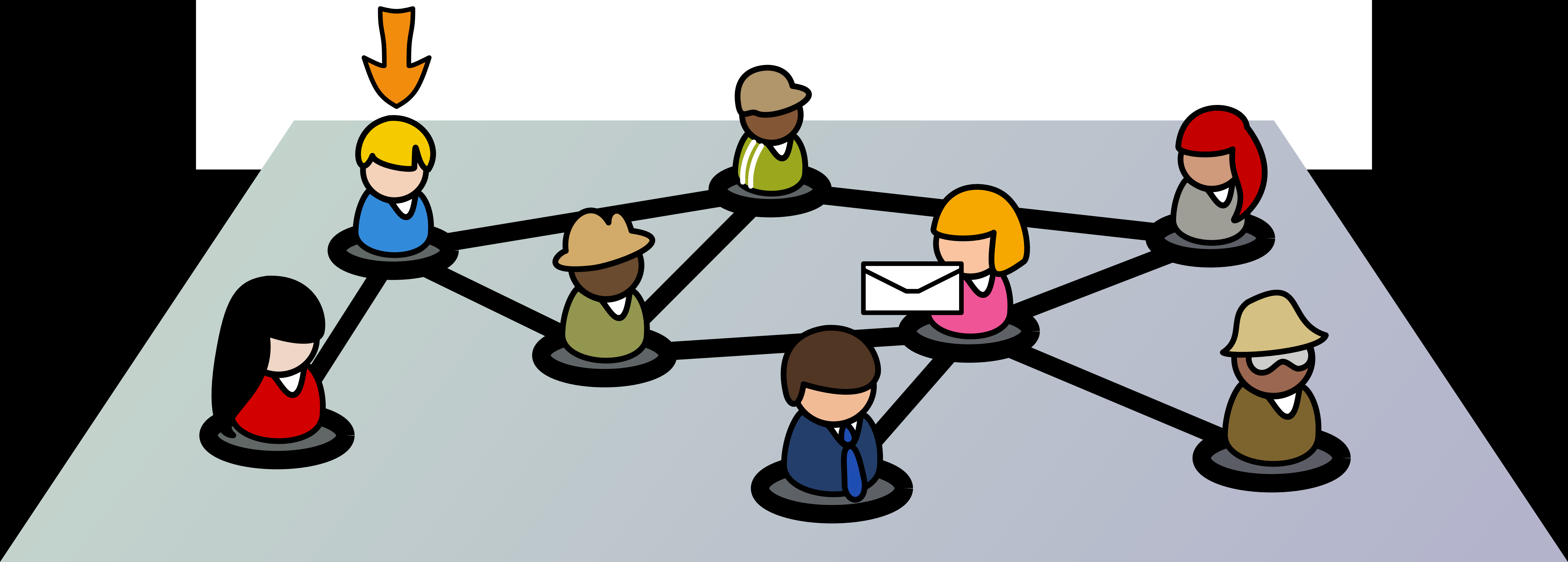
SMALL-WORLD PHENOMENON

- Consider a *social network*
- Milgram experiment [Milgram, 1967]
  - Give letter to random person
  - Select a random target
  - Person should give letter to acquaintence



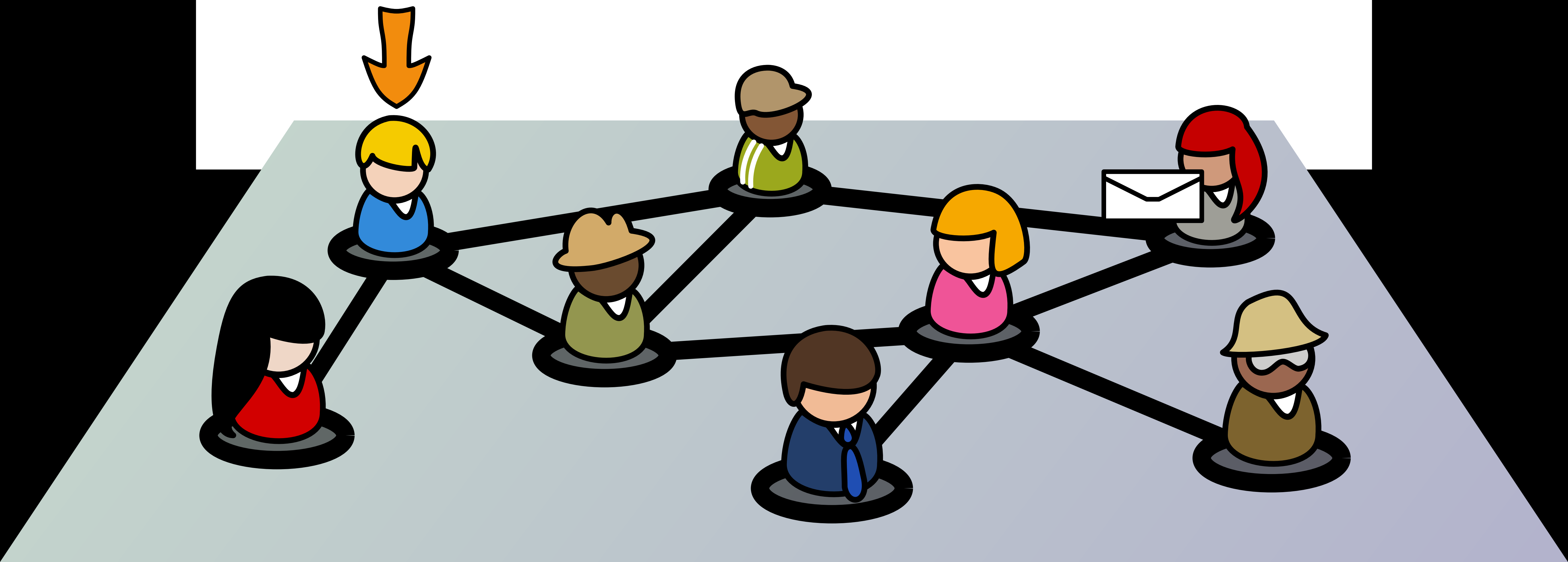
SMALL-WORLD PHENOMENON

- Consider a *social network*
- Milgram experiment [Milgram, 1967]
  - Give letter to random person
  - Select a random target
  - Person should give letter to acquaintence



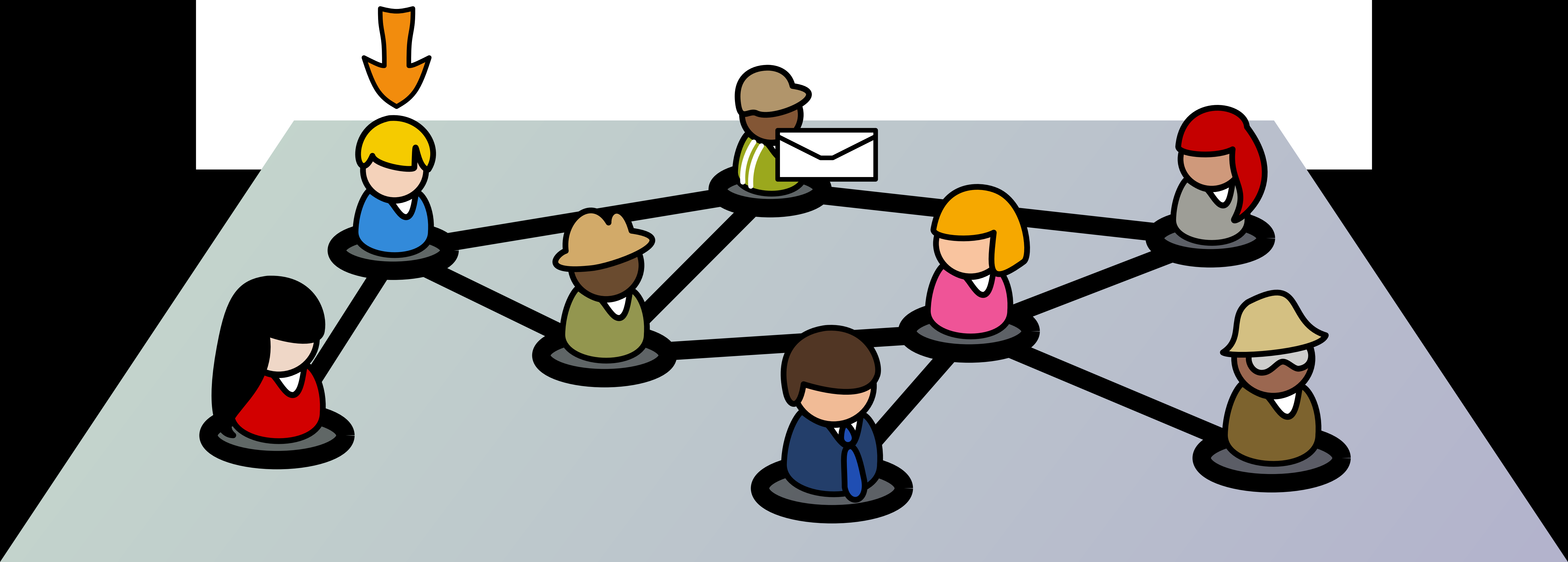
SMALL-WORLD PHENOMENON

- Consider a *social network*
- Milgram experiment [Milgram, 1967]
  - Give letter to random person
  - Select a random target
  - Person should give letter to acquaintence



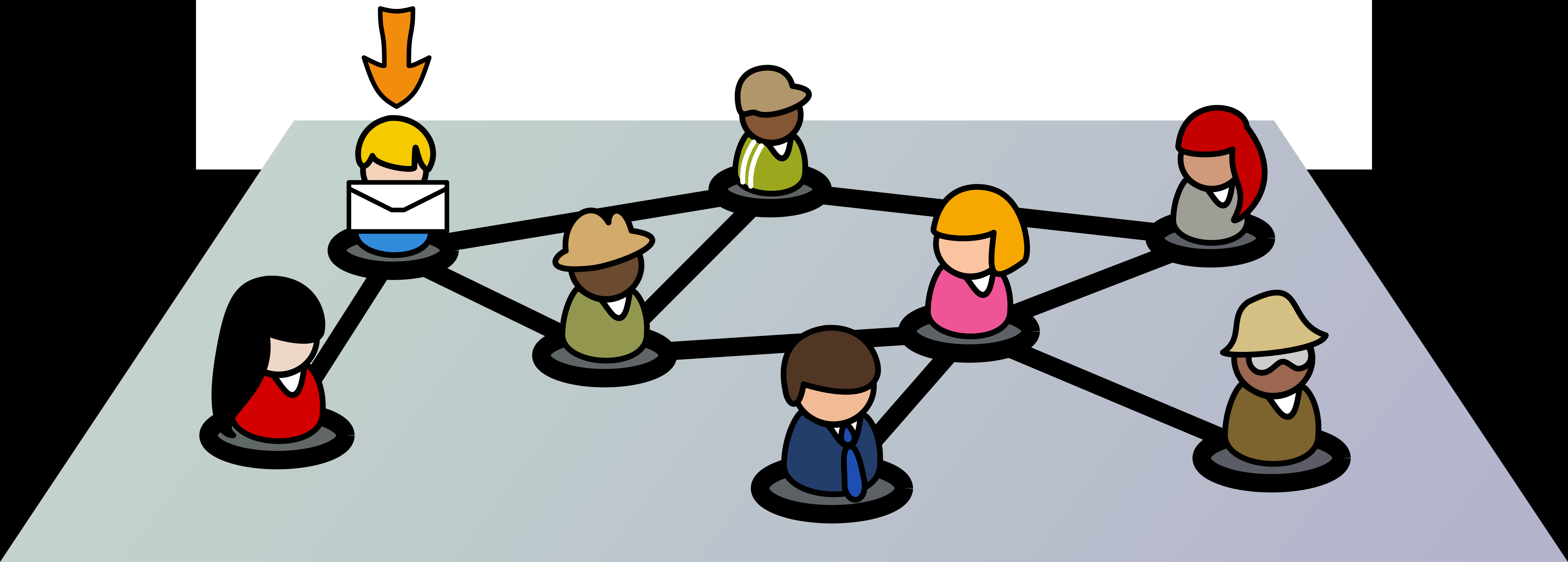
SMALL-WORLD PHENOMENON

- Consider a *social network*
- Milgram experiment [Milgram, 1967]
  - Give letter to random person
  - Select a random target
  - Person should give letter to acquaintence



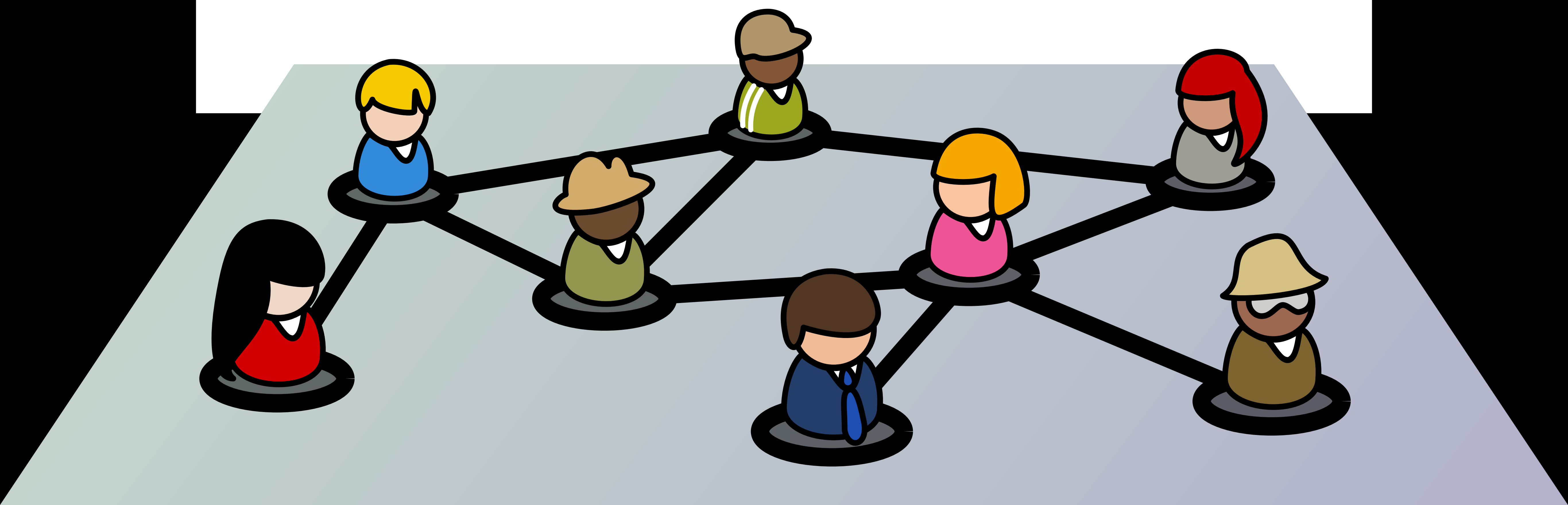
SMALL-WORLD PHENOMENON

- Consider a *social network*
- Milgram experiment [Milgram, 1967]
  - Give letter to random person
  - Select a random target
  - Person should give letter to acquaintence



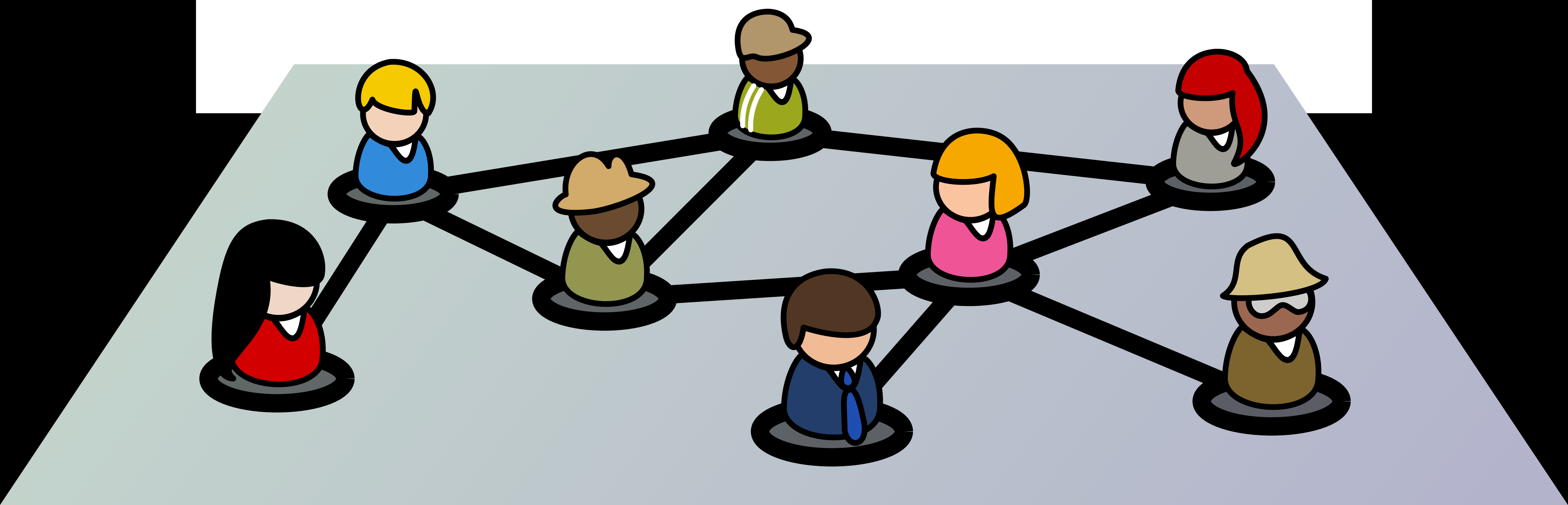
SMALL-WORLD PHENOMENON

- Consider a *social network*
- Milgram experiment [Milgram, 1967]
  - Give letter to random person
  - Select a random target
  - Person should give letter to acquaintence



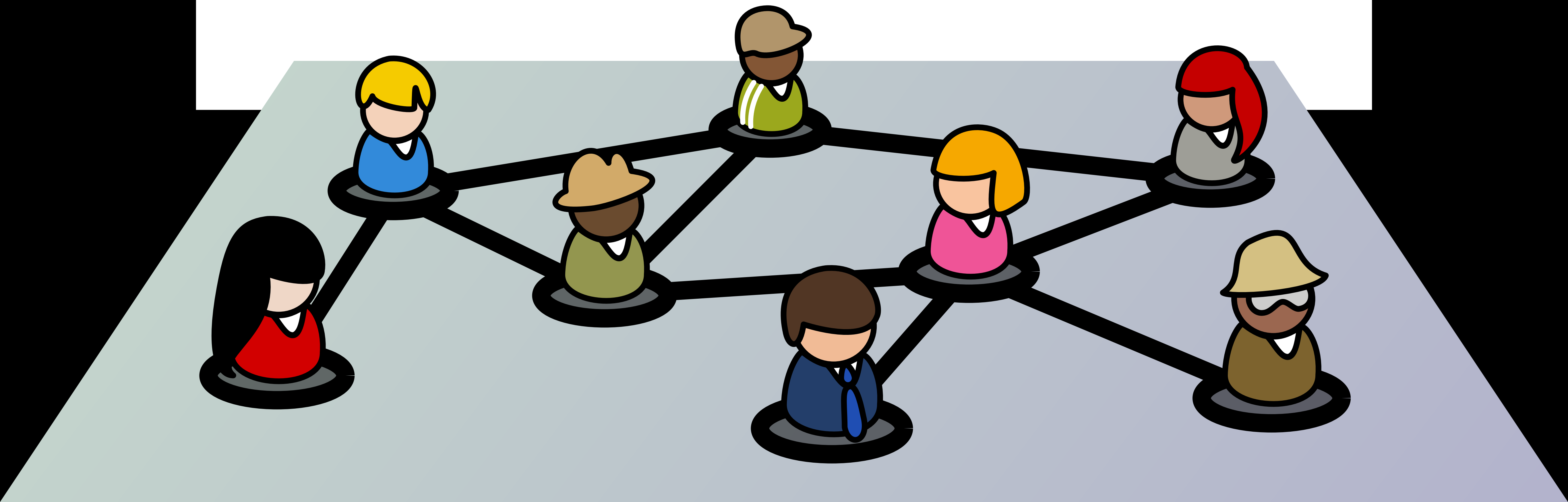
SMALL-WORLD PHENOMENON

- Consider a *social network*
- Milgram experiment [Milgram, 1967]
  - Give letter to random person
  - Select a random target
  - Person should give letter to acquaintence
- Conclusions



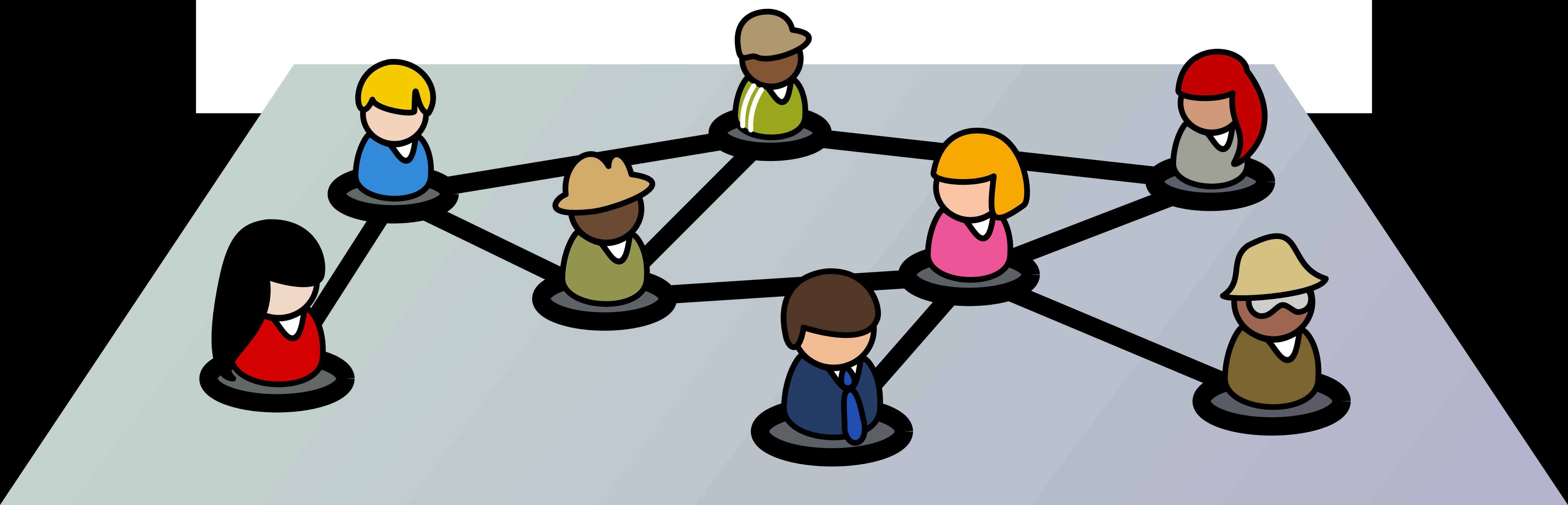
SMALL-WORLD PHENOMENON

- Consider a *social network*
- Milgram experiment [Milgram, 1967]
  - Give letter to random person
  - Select a random target
  - Person should give letter to acquaintence
- Conclusions
  - Short paths exist between all people



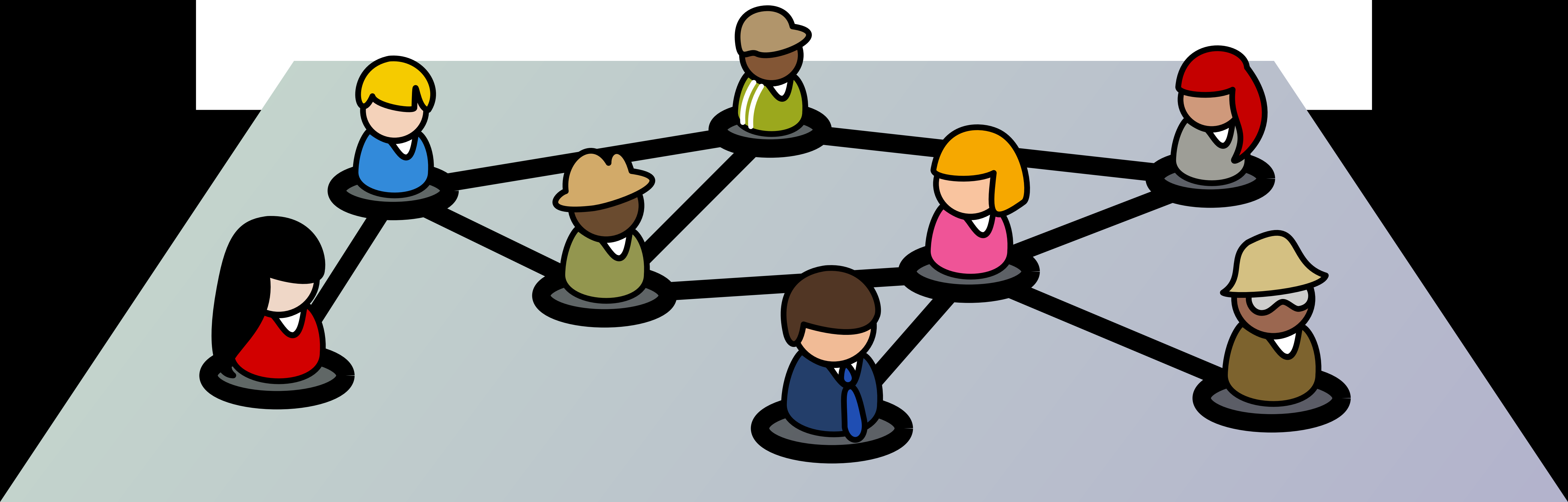
SMALL-WORLD PHENOMENON

- Consider a *social network*
- Milgram experiment [Milgram, 1967]
  - Give letter to random person
  - Select a random target
  - Person should give letter to acquaintence
- Conclusions
  - Short paths exist between all people
  - “six degrees of separation”

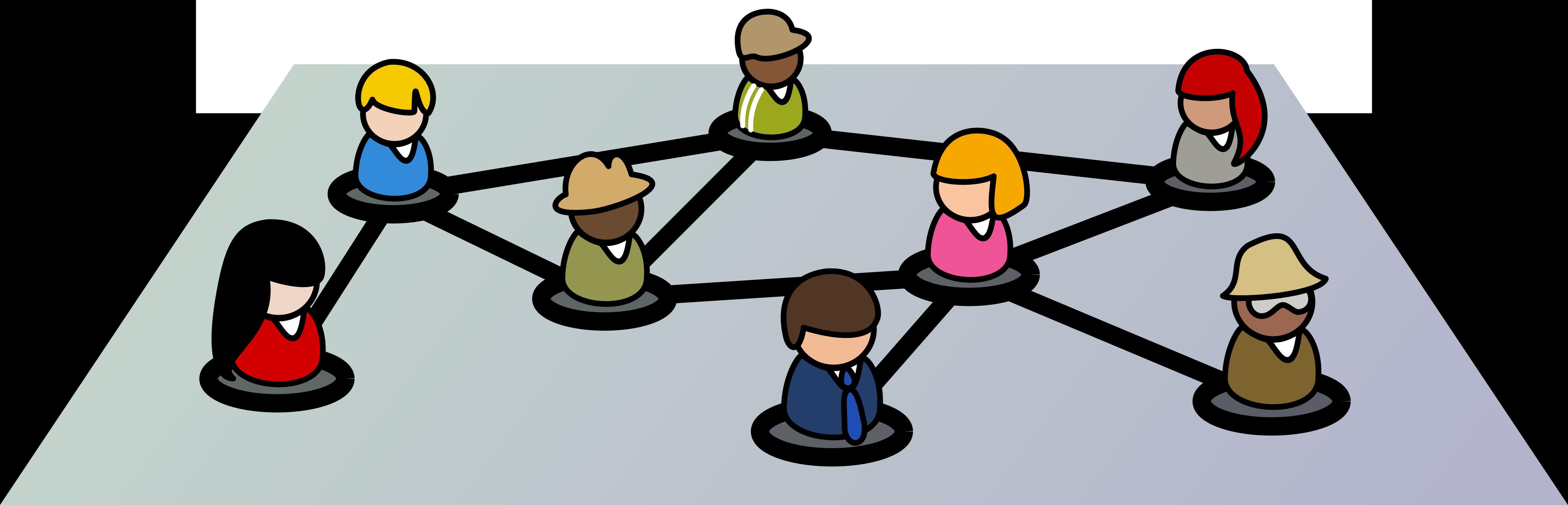


SMALL-WORLD PHENOMENON

- Consider a *social network*
- Milgram experiment [Milgram, 1967]
  - Give letter to random person
  - Select a random target
  - Person should give letter to acquaintence
- Conclusions
  - Short paths exist between all people
  - "six degrees of separation"
  - ... and people are able to *find* these paths

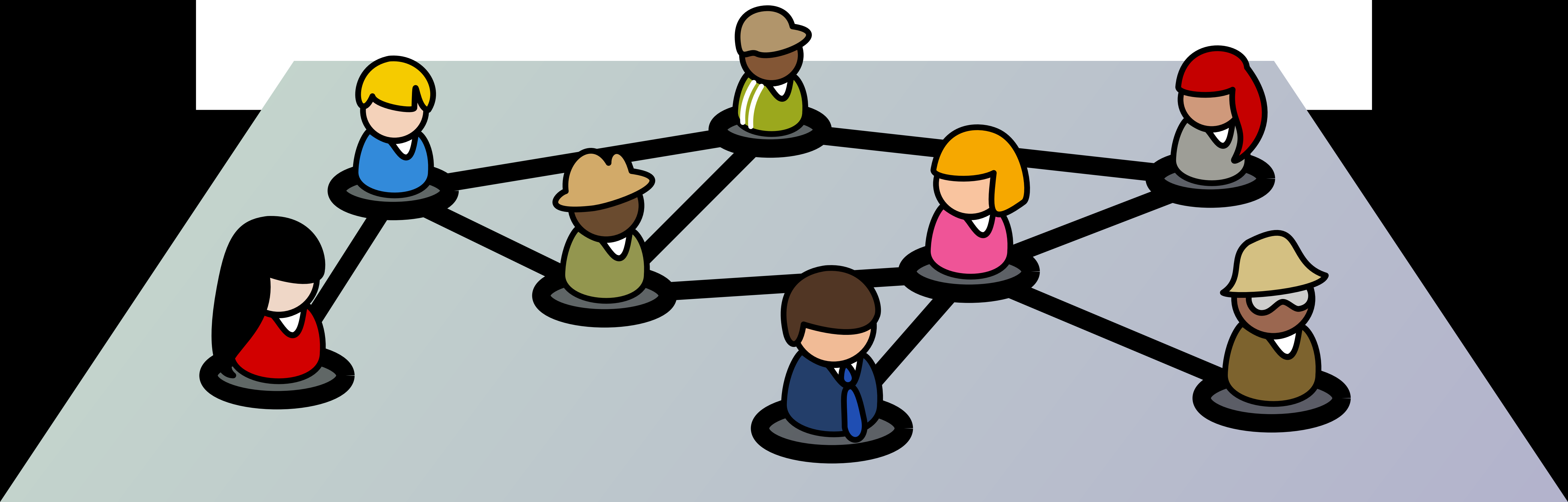


ROUTING IN A SMALL WORLD



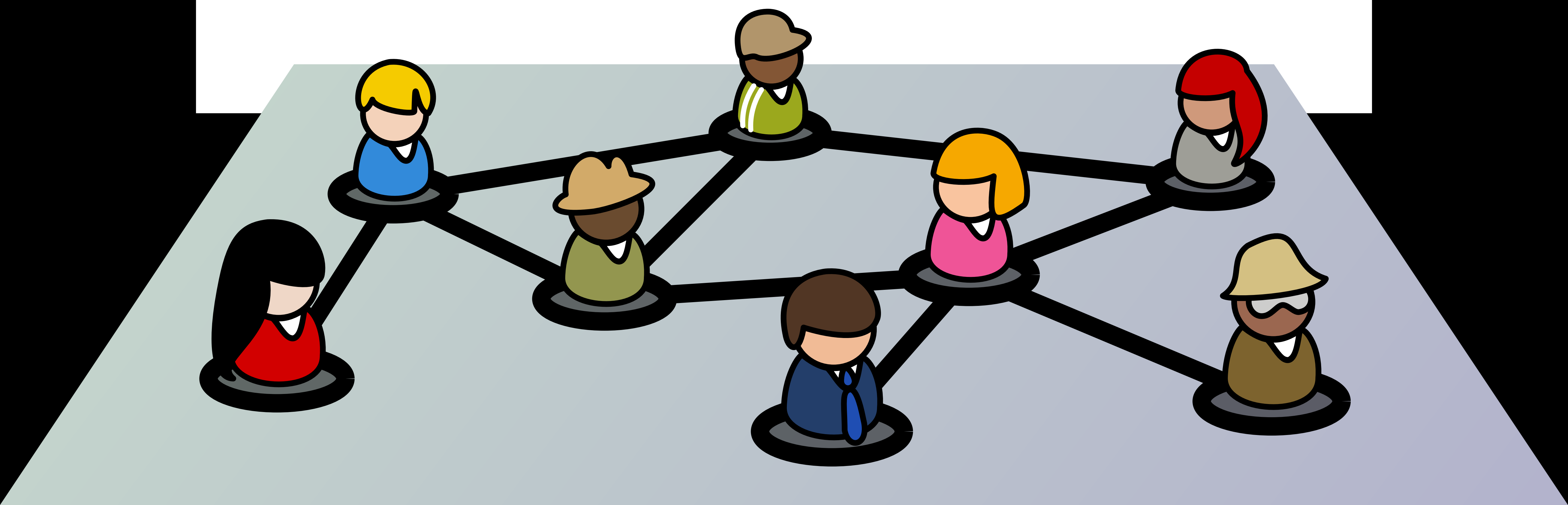
ROUTING IN A SMALL WORLD

- Follow-up experiments



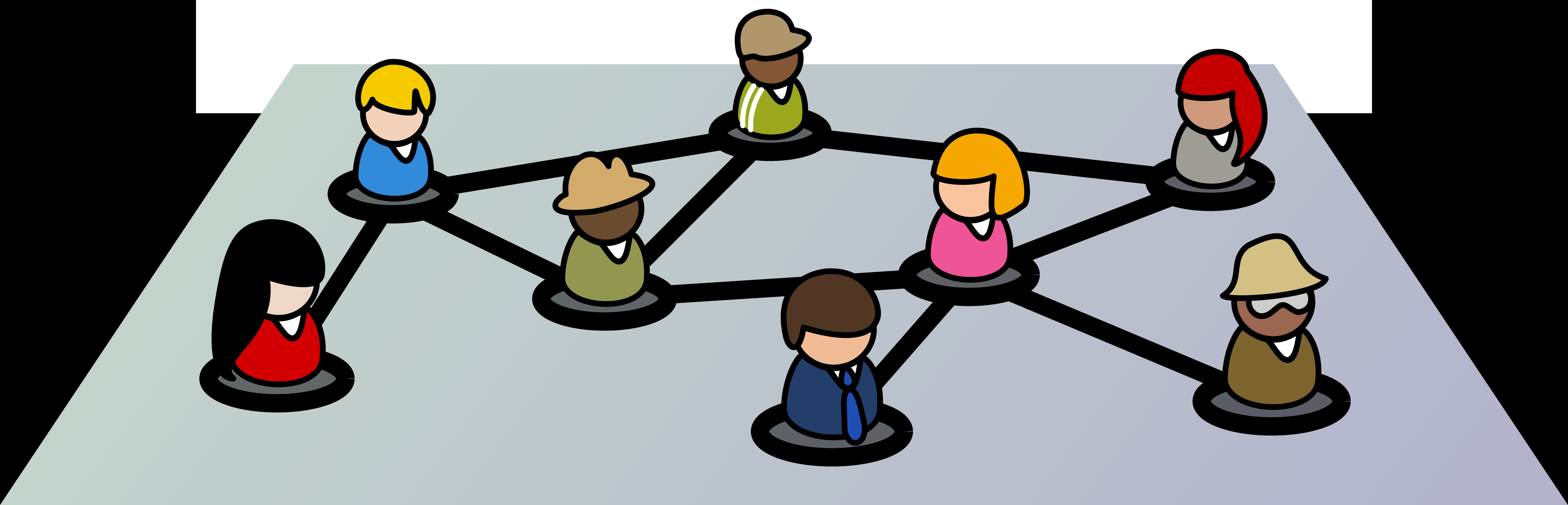
ROUTING IN A SMALL WORLD

- Follow-up experiments
  - Routes between people with the same occupation are shorter [Hunter & Shotland, 1974]



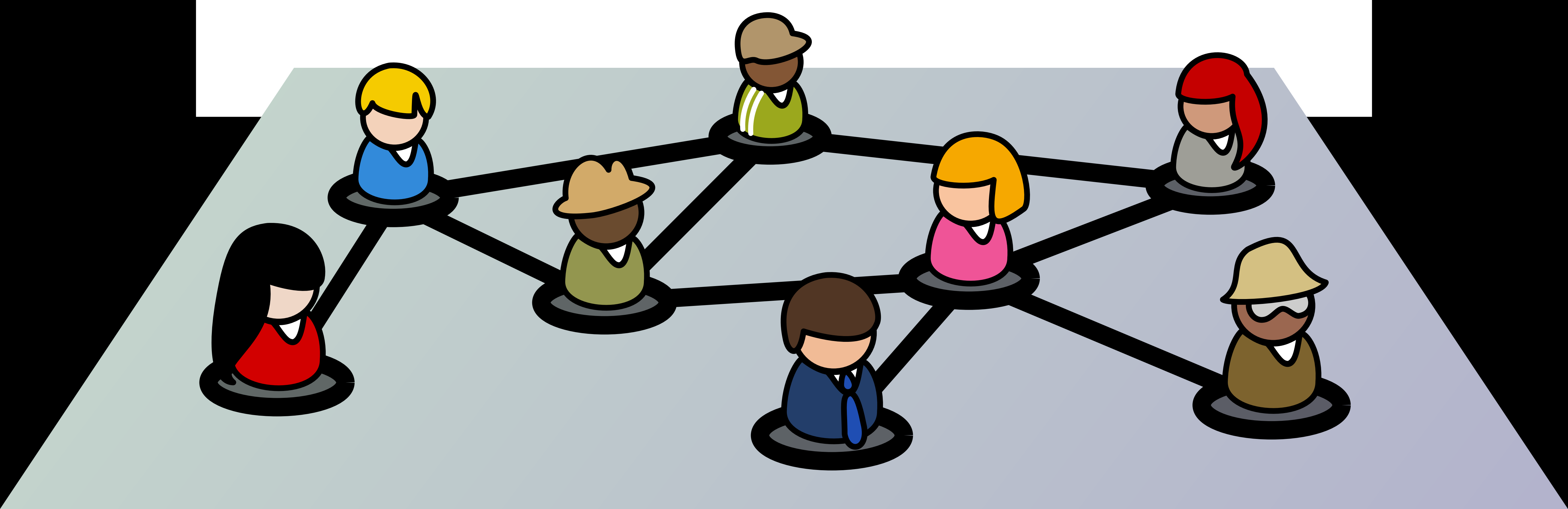
ROUTING IN A SMALL WORLD

- Follow-up experiments
  - Routes between people with the same occupation are shorter [Hunter & Shotland, 1974]
  - Decisions who to route to are mostly "categorical" in nature [Killworth & Bernard, 1978]



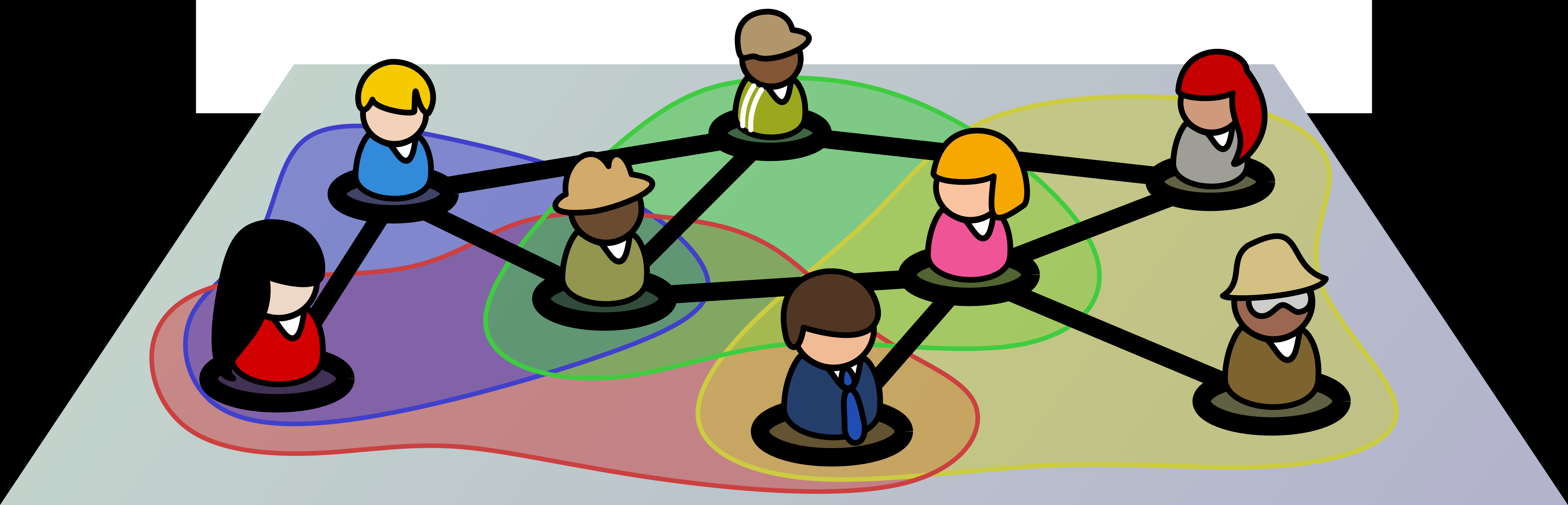
ROUTING IN A SMALL WORLD

- Follow-up experiments
  - Routes between people with the same occupation are shorter [Hunter & Shotland, 1974]
  - Decisions who to route to are mostly "categorical" in nature [Killworth & Bernard, 1978]
- Consider a set of *categories*



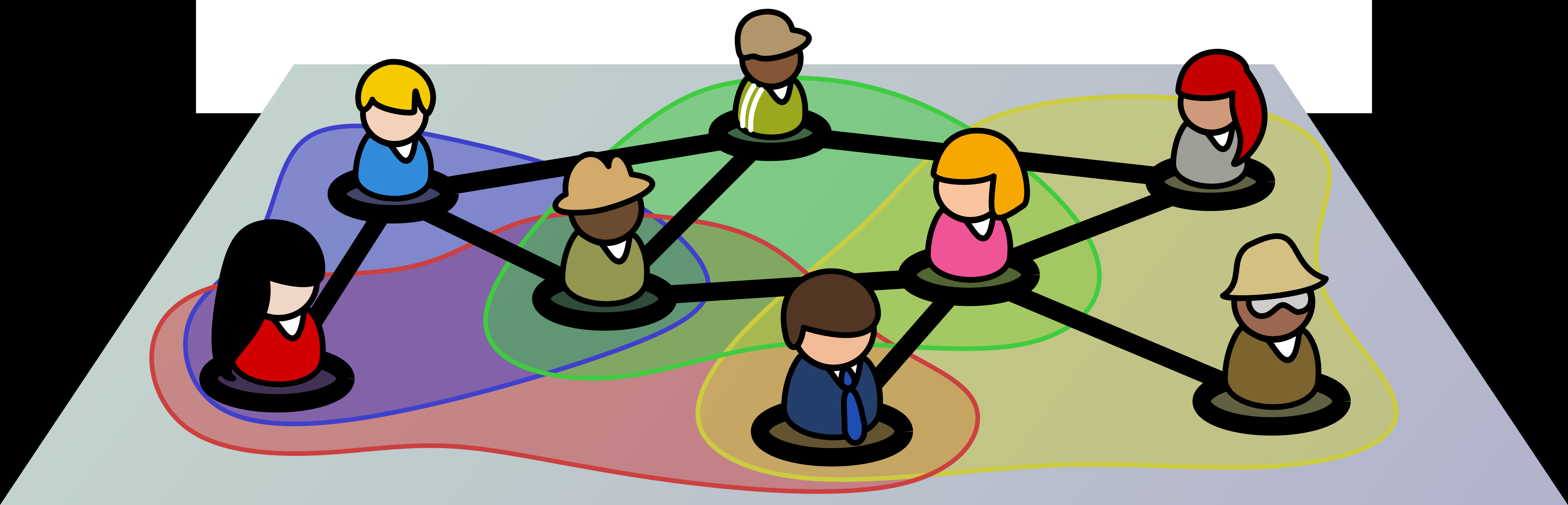
ROUTING IN A SMALL WORLD

- Follow-up experiments
  - Routes between people with the same occupation are shorter [Hunter & Shotland, 1974]
  - Decisions who to route to are mostly "categorical" in nature [Killworth & Bernard, 1978]
- Consider a set of *categories*



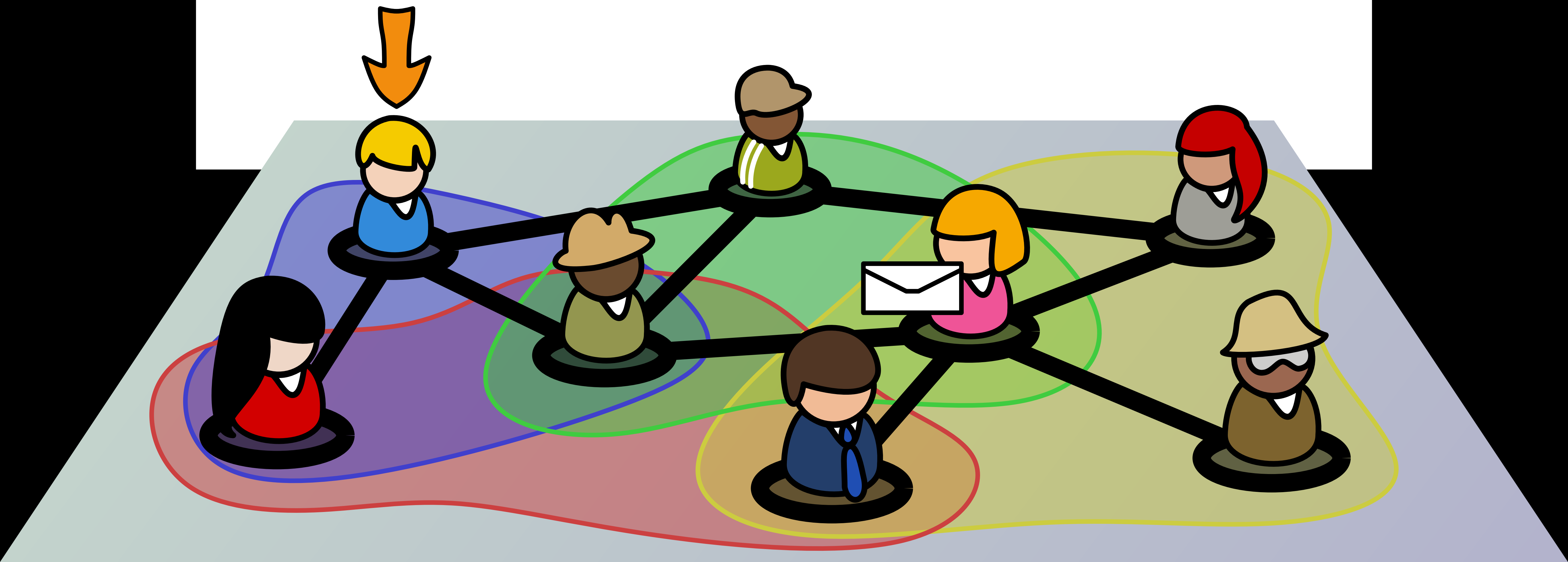
ROUTING IN A SMALL WORLD

- Follow-up experiments
  - Routes between people with the same occupation are shorter [Hunter & Shotland, 1974]
  - Decisions who to route to are mostly "categorical" in nature [Killworth & Bernard, 1978]
- Consider a set of *categories*
- Make decisions based on them



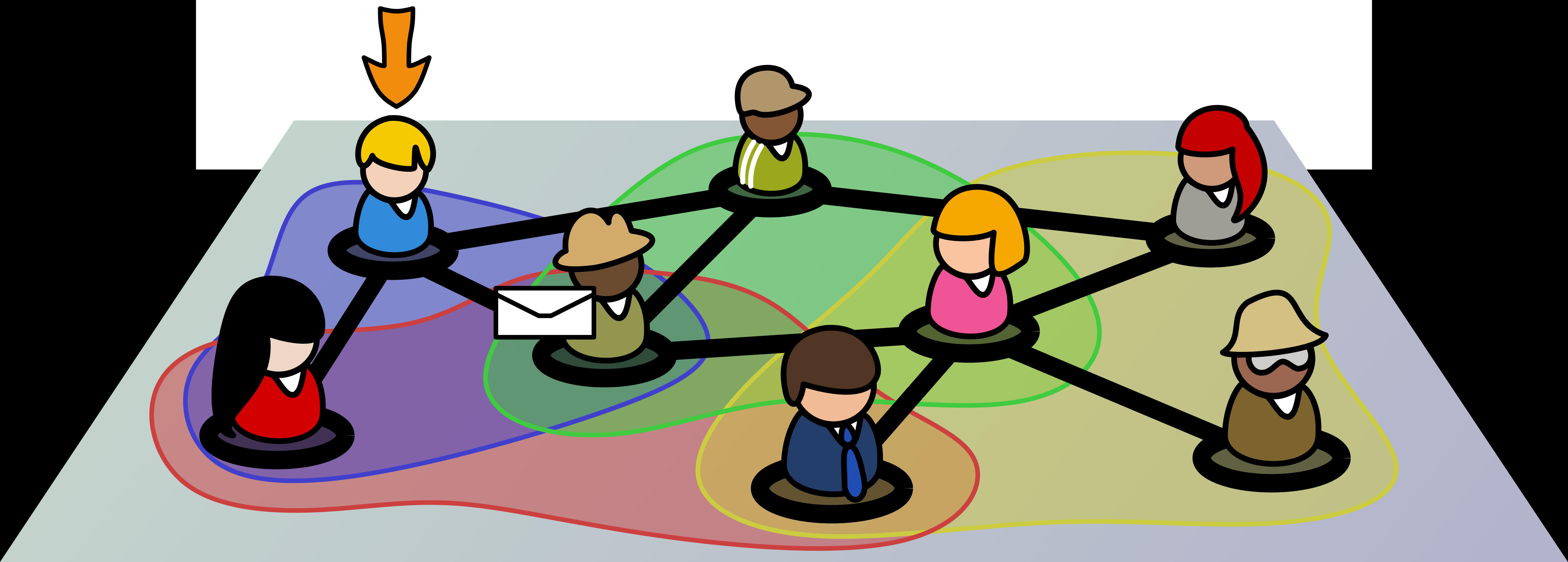
ROUTING IN A SMALL WORLD

- Follow-up experiments
  - Routes between people with the same occupation are shorter [Hunter & Shotland, 1974]
  - Decisions who to route to are mostly "categorical" in nature [Killworth & Bernard, 1978]
- Consider a set of *categories*
- Make decisions based on them



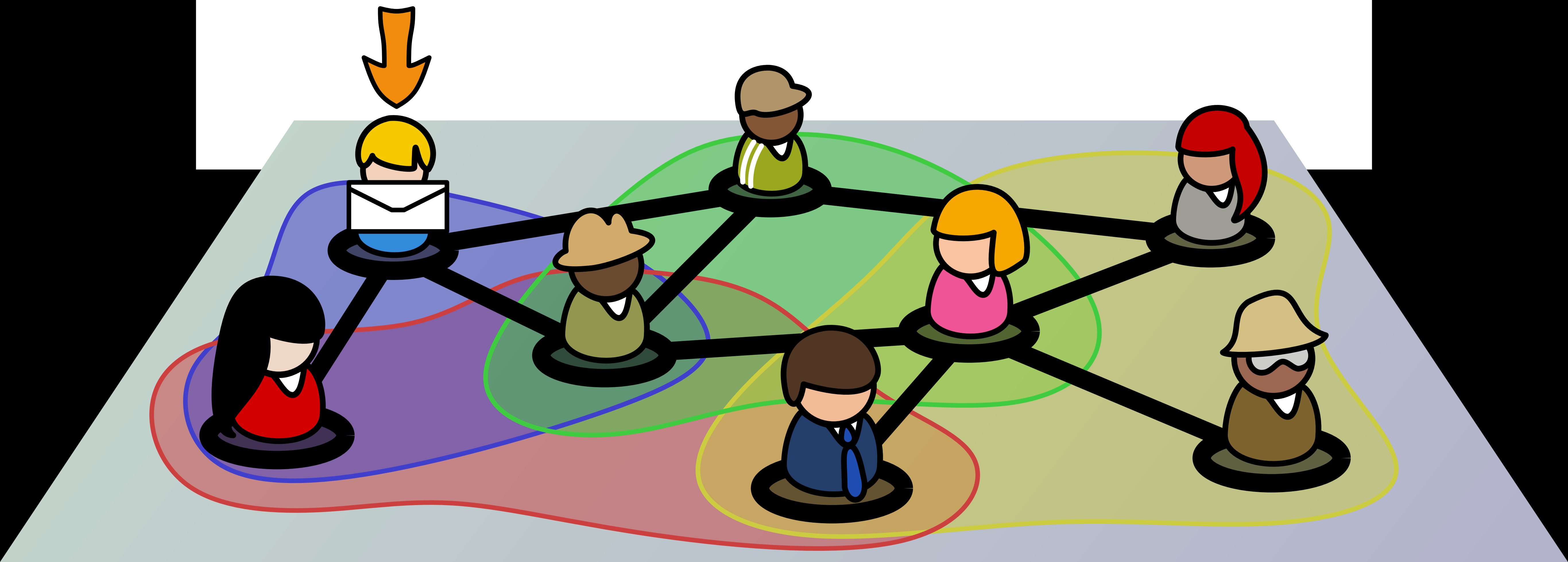
ROUTING IN A SMALL WORLD

- Follow-up experiments
  - Routes between people with the same occupation are shorter [Hunter & Shotland, 1974]
  - Decisions who to route to are mostly "categorical" in nature [Killworth & Bernard, 1978]
- Consider a set of *categories*
- Make decisions based on them



ROUTING IN A SMALL WORLD

- Follow-up experiments
  - Routes between people with the same occupation are shorter [Hunter & Shotland, 1974]
  - Decisions who to route to are mostly "categorical" in nature [Killworth & Bernard, 1978]
- Consider a set of *categories*
- Make decisions based on them



MAIN QUESTIONS

MAIN QUESTIONS

- Is this feasible?

MAIN QUESTIONS

- Is this feasible?
  - Assume people use a simple category-based routing algorithm

MAIN QUESTIONS

- Is this feasible?
  - Assume people use a simple category-based routing algorithm
  - Under what conditions of a network and set of categories does simple routing work?

MAIN QUESTIONS

- Is this feasible?
  - Assume people use a simple category-based routing algorithm
  - Under what conditions of a network and set of categories does simple routing work?
  - How much does an individual need to know for this to work?

PART II
DEFINITIONS & RESULTS

SOME NOTATION

SOME NOTATION

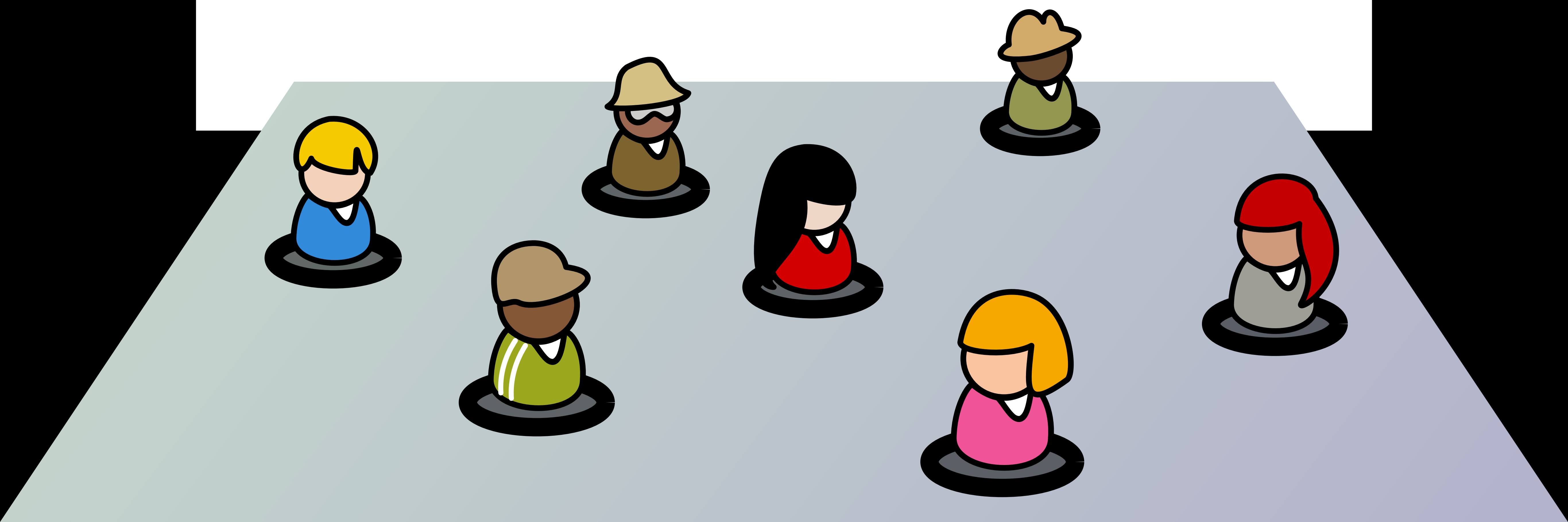
- Ingredients

SOME NOTATION

- Ingredients
  - A set U of n objects

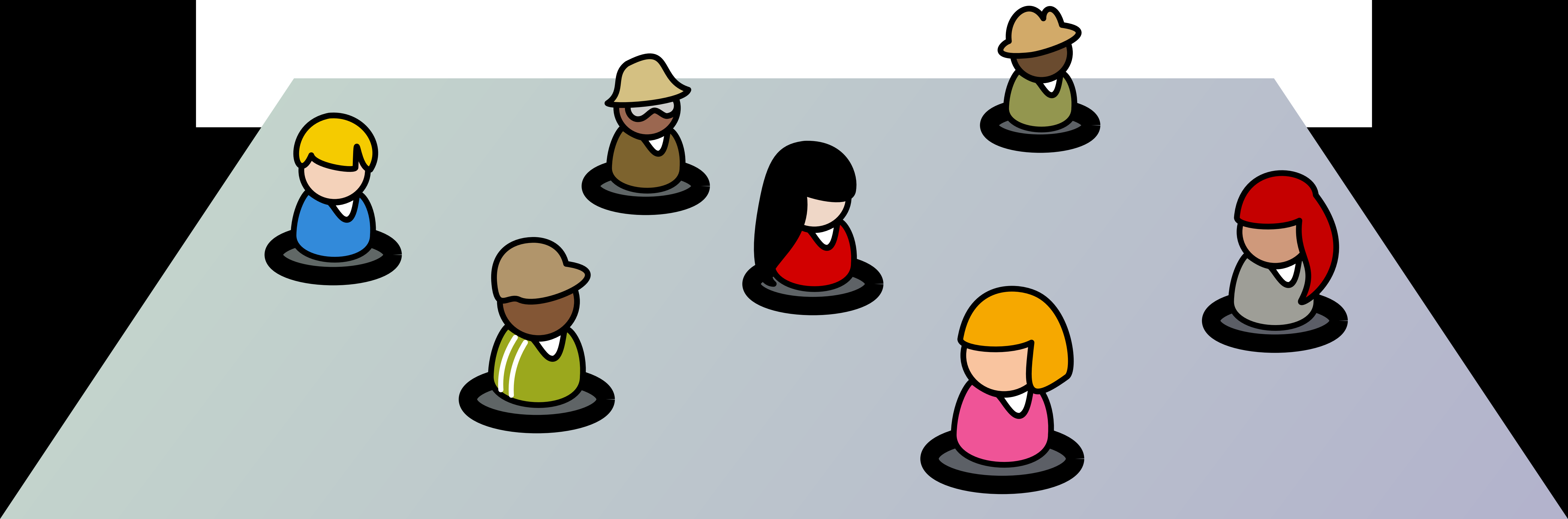
SOME NOTATION

- Ingredients
  - A set U of n objects



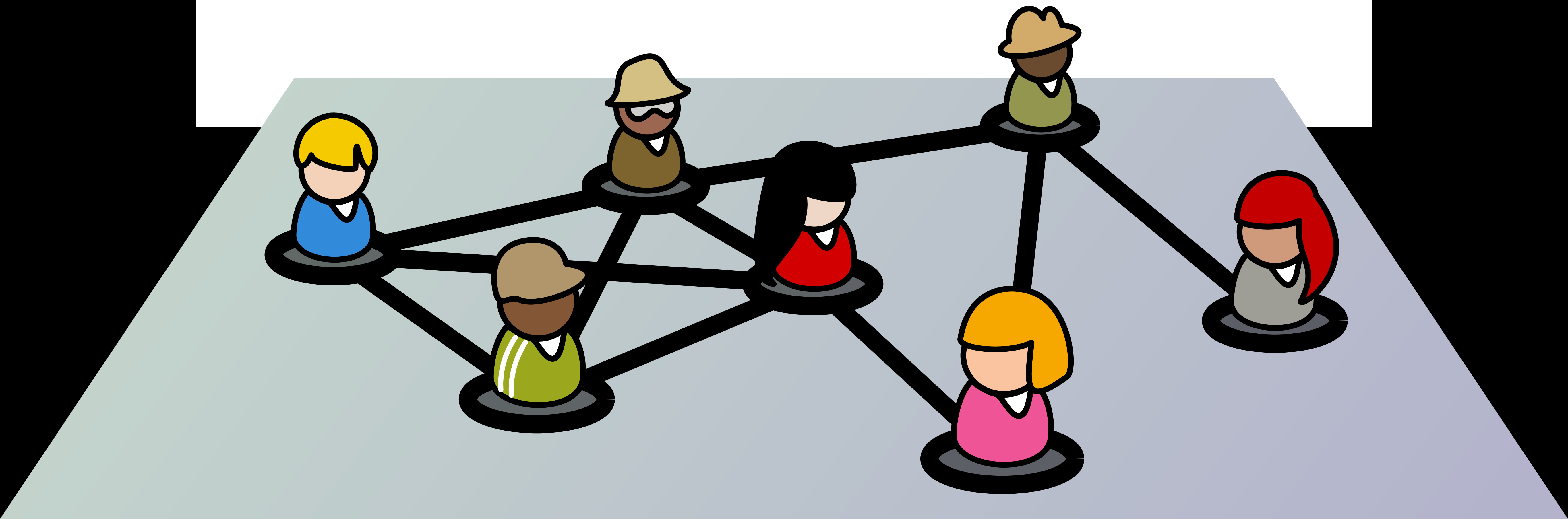
SOME NOTATION

- Ingredients
  - A set U of n objects
  - A set E of relations, resulting in a graph $G = (U, E)$



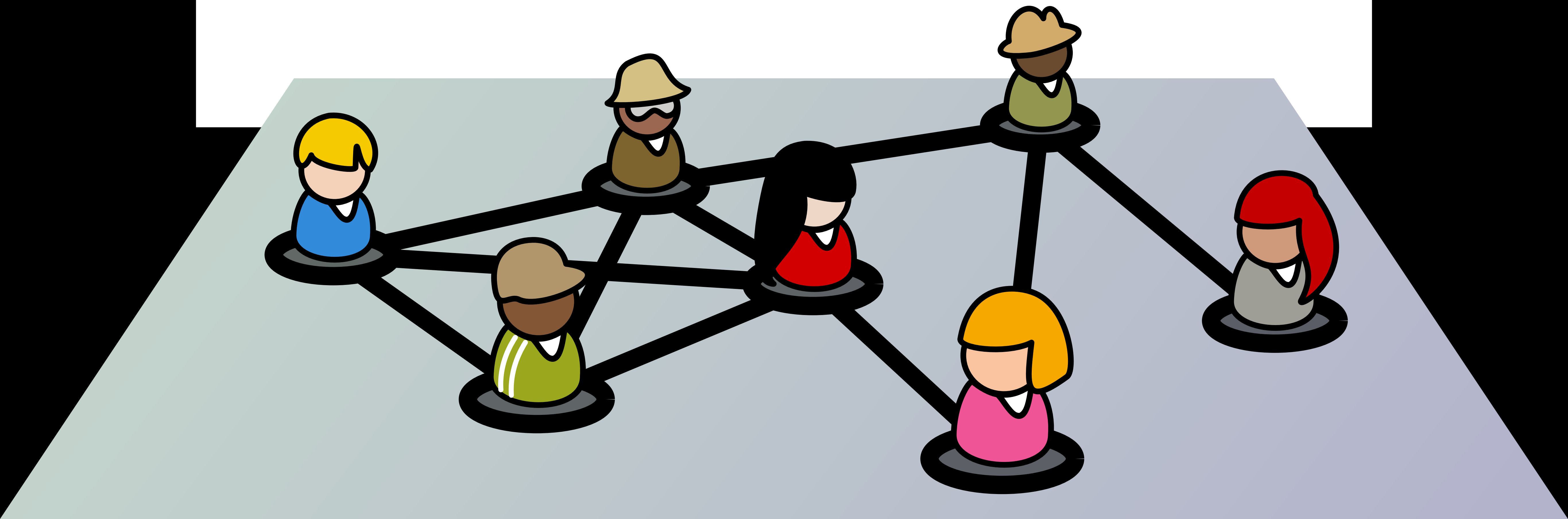
SOME NOTATION

- Ingredients
  - A set U of n objects
  - A set E of relations, resulting in a graph $G = (U, E)$



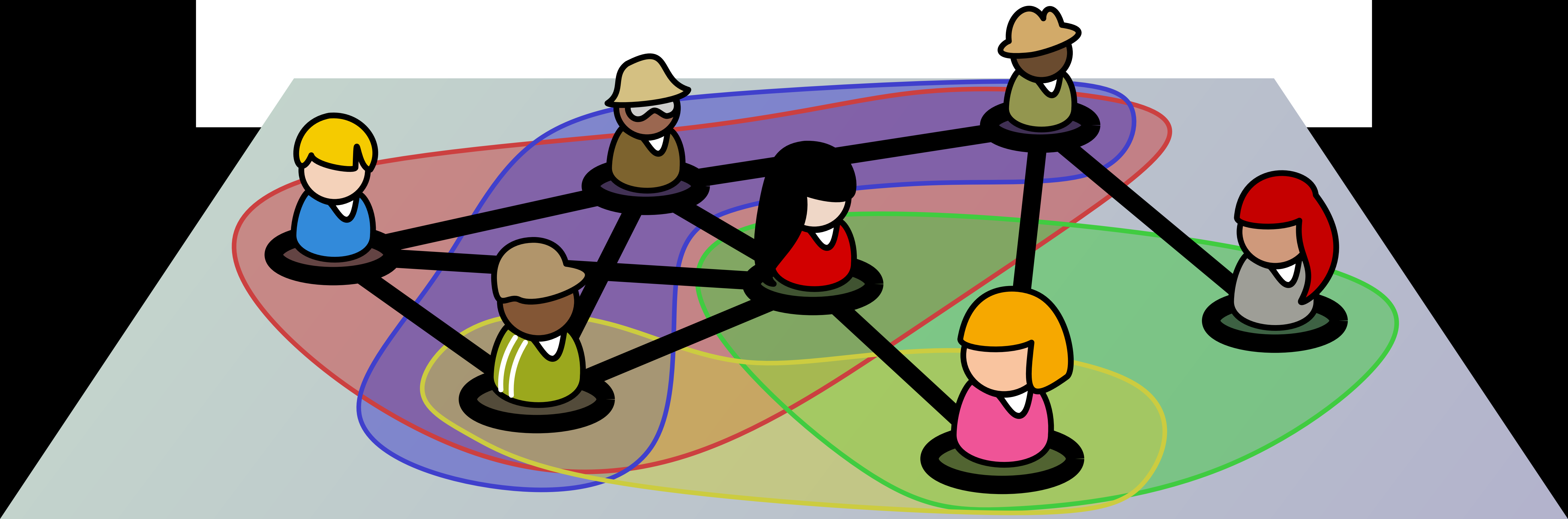
SOME NOTATION

- Ingredients
  - A set U of n objects
  - A set E of relations, resulting in a graph $G = (U, E)$
  - A set $\mathcal{S} \subseteq 2^U$ of categories



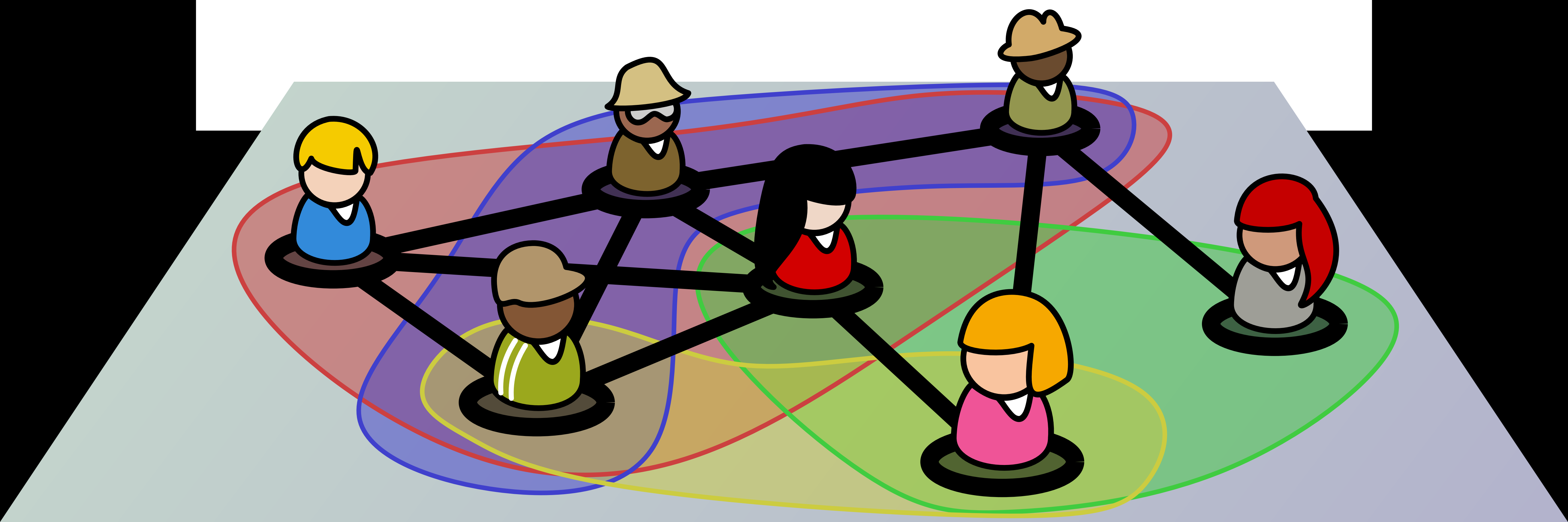
SOME NOTATION

- Ingredients
  - A set U of n objects
  - A set E of relations, resulting in a graph $G = (U, E)$
  - A set $\mathcal{S} \subseteq 2^U$ of categories



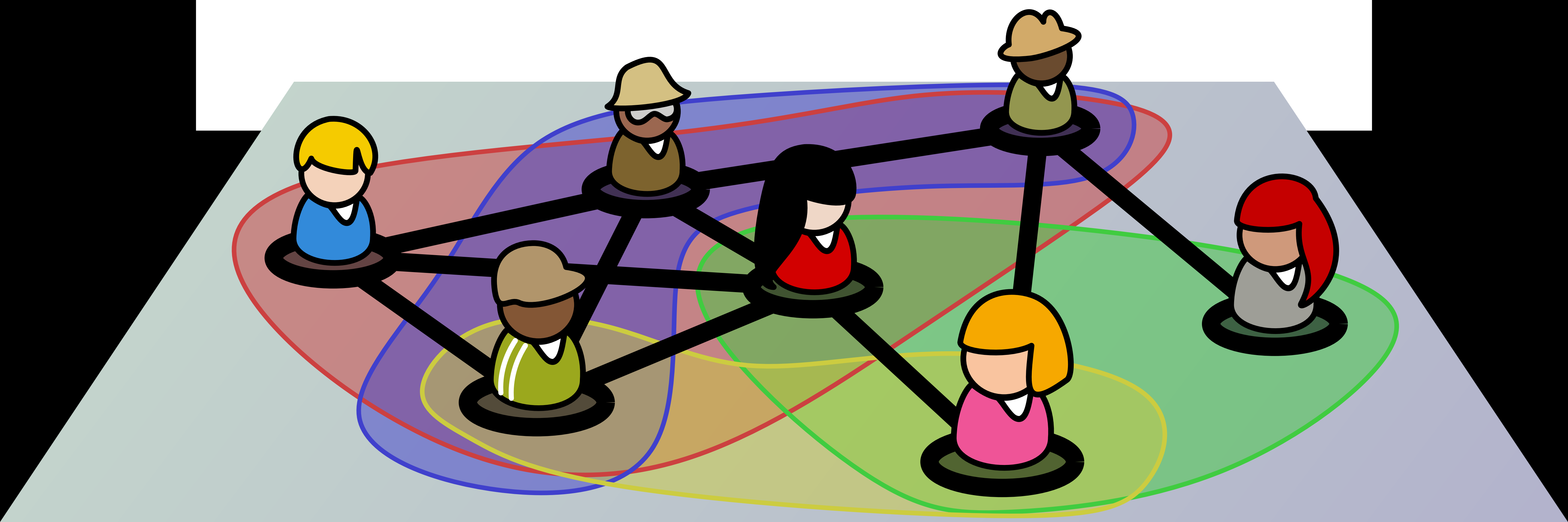
SOME NOTATION

- Ingredients
  - A set U of n objects
  - A set E of relations, resulting in a graph $G = (U, E)$
  - A set $\mathcal{S} \subseteq 2^U$ of categories
- Basic concepts



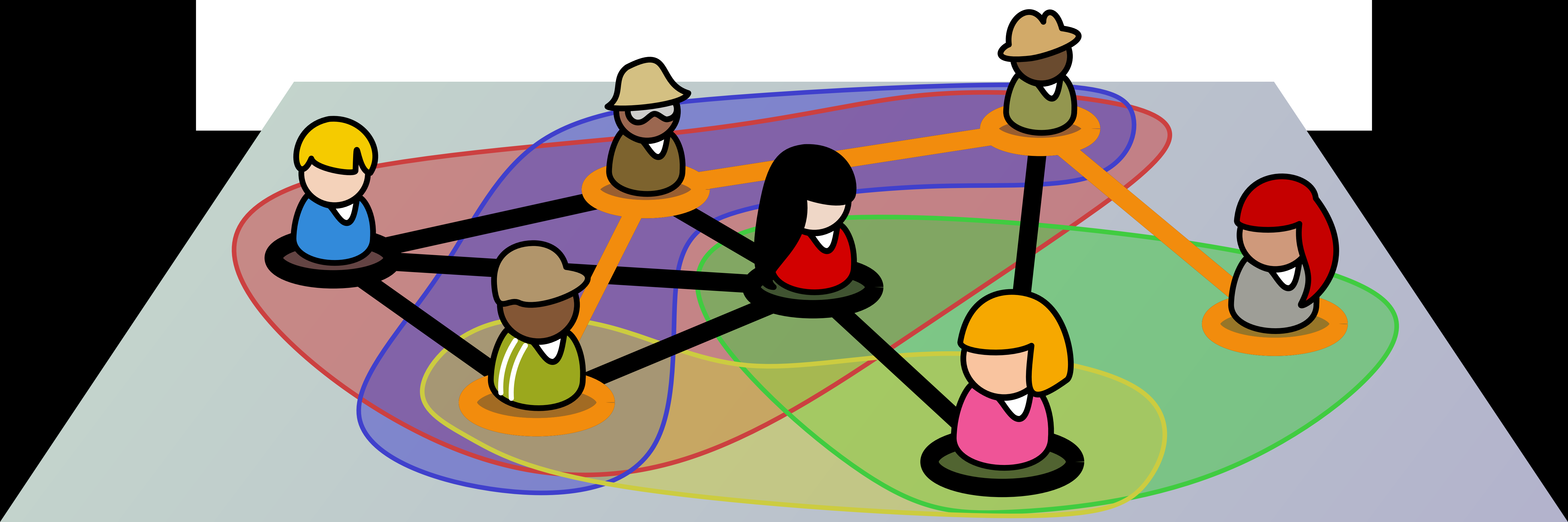
SOME NOTATION

- Ingredients
  - A set U of n objects
  - A set E of relations, resulting in a graph $G = (U, E)$
  - A set $\mathcal{S} \subseteq 2^U$ of categories
- Basic concepts
  - $\mathrm{diam}(G)$: longest shortest path in G



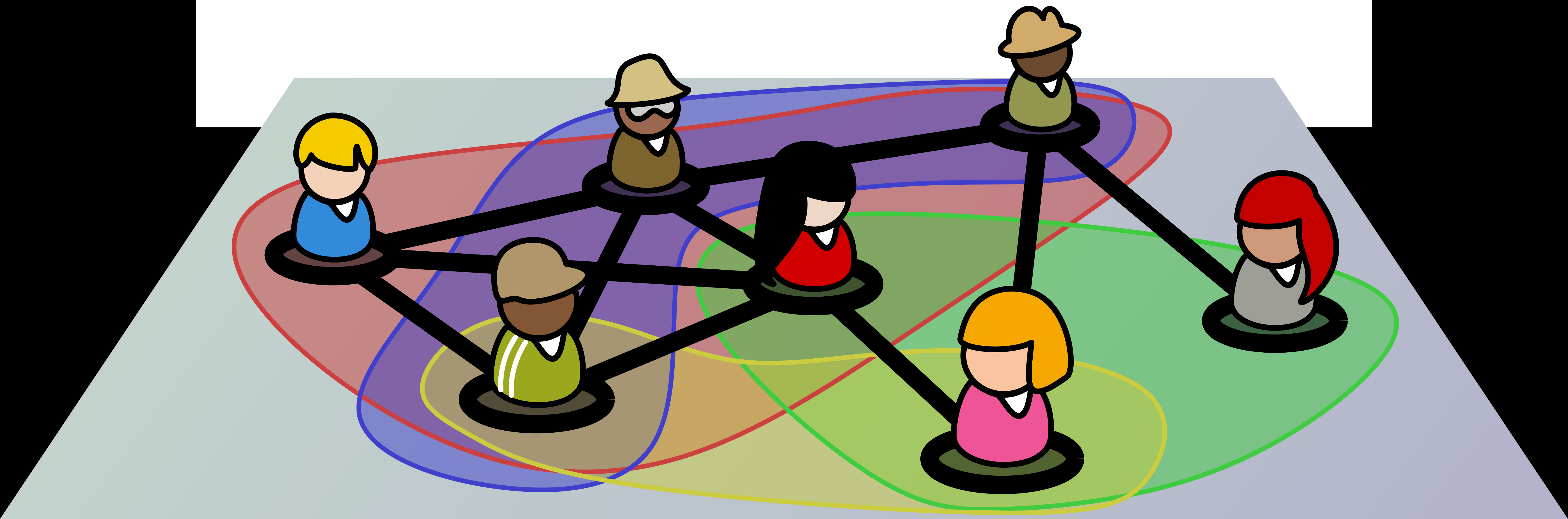
SOME NOTATION

- Ingredients
  - A set U of n objects
  - A set E of relations, resulting in a graph $G = (U, E)$
  - A set $\mathcal{S} \subseteq 2^U$ of categories
- Basic concepts
  - $\mathrm{diam}(G)$: longest shortest path in G



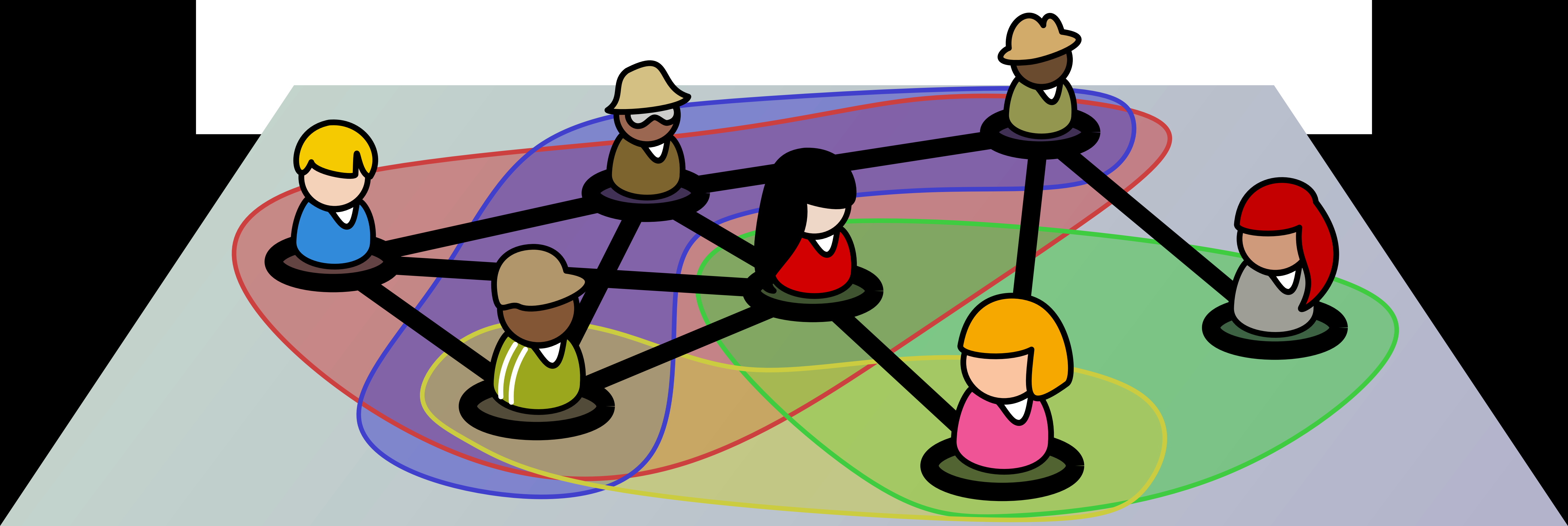
SOME NOTATION

- Ingredients
  - A set U of n objects
  - A set E of relations, resulting in a graph $G = (U, E)$
  - A set $\mathcal{S} \subseteq 2^U$ of categories
- Basic concepts
  - $\mathrm{diam}(G)$: longest shortest path in G



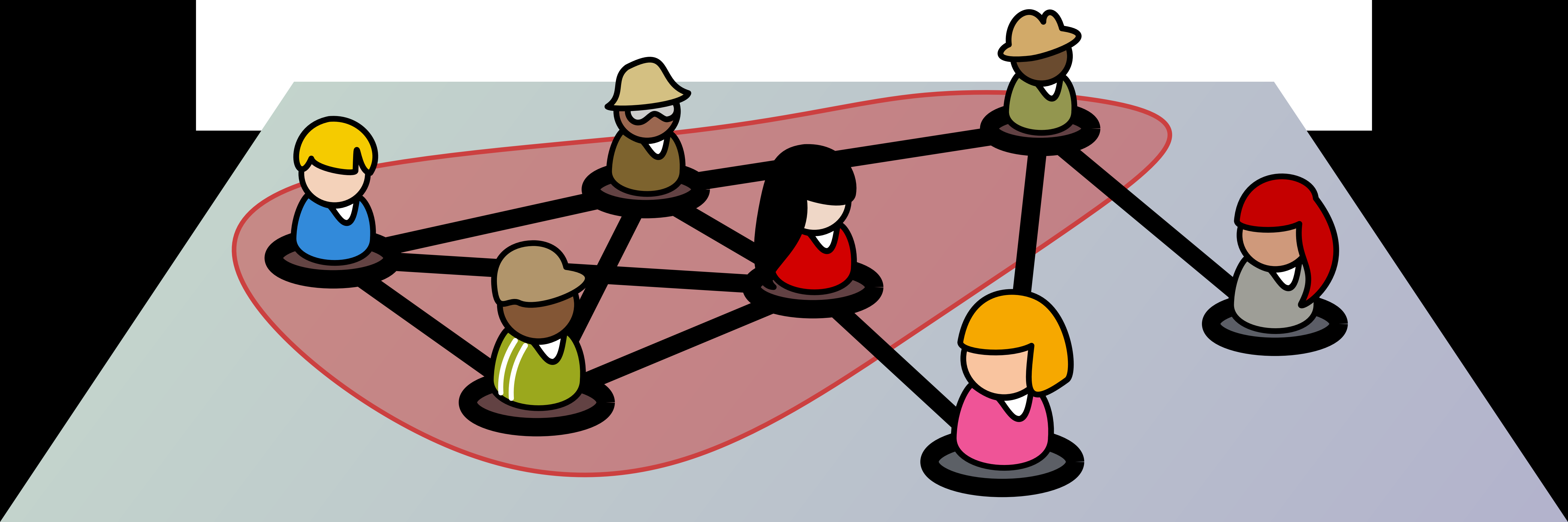
SOME NOTATION

- Ingredients
  - A set U of n objects
  - A set E of relations, resulting in a graph $G = (U, E)$
  - A set $\mathcal{S} \subseteq 2^U$ of categories
- Basic concepts
  - $\mathrm{diam}(G)$: longest shortest path in G
  - $G[C]$: subgraph induced by C



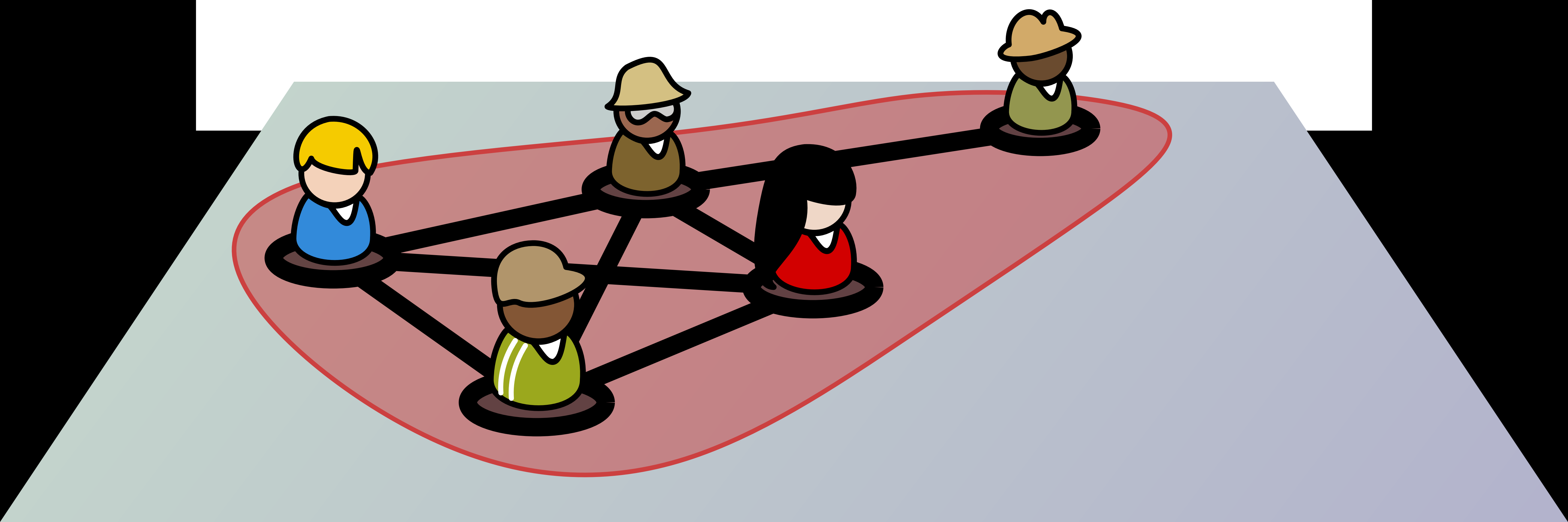
SOME NOTATION

- Ingredients
  - A set U of n objects
  - A set E of relations, resulting in a graph $G = (U, E)$
  - A set $\mathcal{S} \subseteq 2^U$ of categories
- Basic concepts
  - $\mathrm{diam}(G)$: longest shortest path in G
  - $G[C]$: subgraph induced by C



SOME NOTATION

- Ingredients
  - A set U of n objects
  - A set E of relations, resulting in a graph $G = (U, E)$
  - A set $\mathcal{S} \subseteq 2^U$ of categories
- Basic concepts
  - $\mathrm{diam}(G)$: longest shortest path in G
  - $G[C]$: subgraph induced by C



SIMPLE ROUTING

SIMPLE ROUTING

- Algorithm

SIMPLE ROUTING

- Algorithm
  - Give message to neighbour who shares most categories with the target

SIMPLE ROUTING

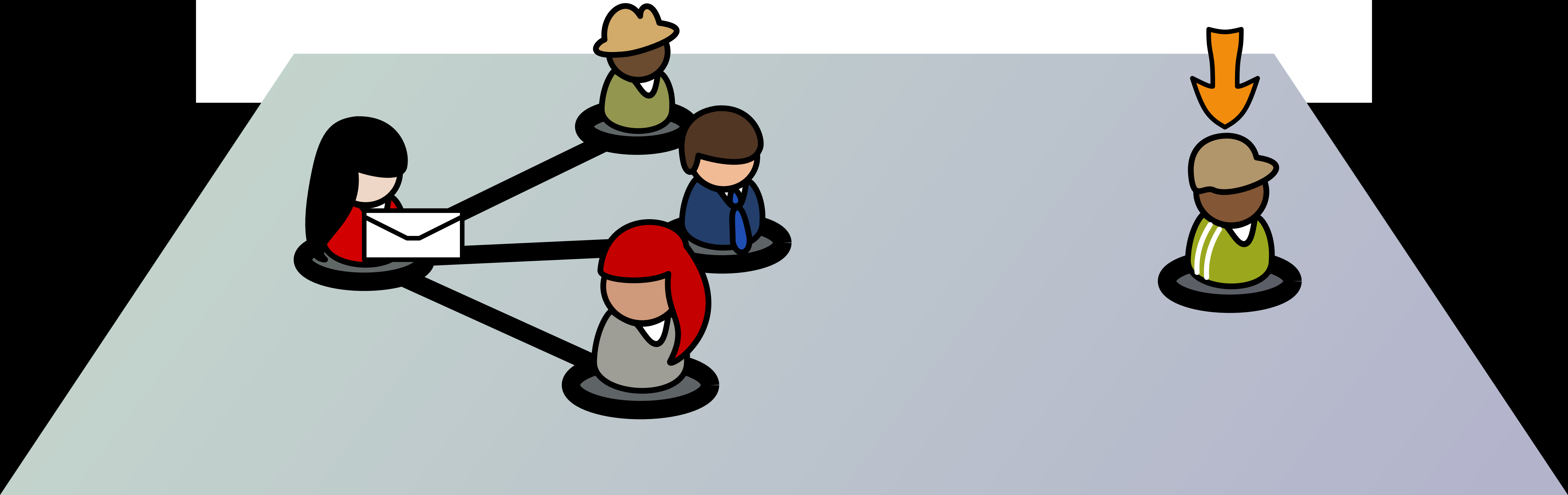
- Algorithm
  - Give message to neighbour who shares most categories with the target
  - $\max_{u:(s,u)\in E} |\{C : s \notin C \wedge u, t \in C\}|$

SIMPLE ROUTING

- Algorithm
  - Give message to neighbour who shares most categories with the target
  - $\max_{u:(s,u)\in E} |\{C : s \notin C \wedge u, t \in C\}|$

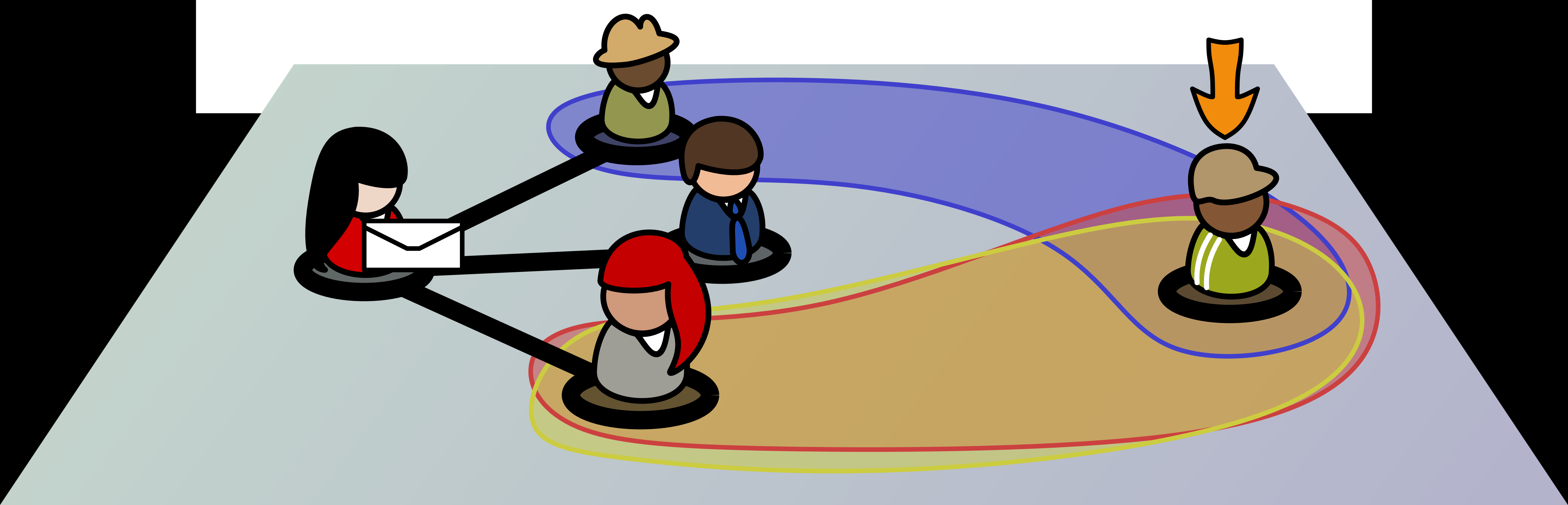
SIMPLE ROUTING

- Algorithm
  - Give message to neighbour who shares most categories with the target
  - $\max_{u:(s,u)\in E} |\{C : s \notin C \wedge u, t \in C\}|$



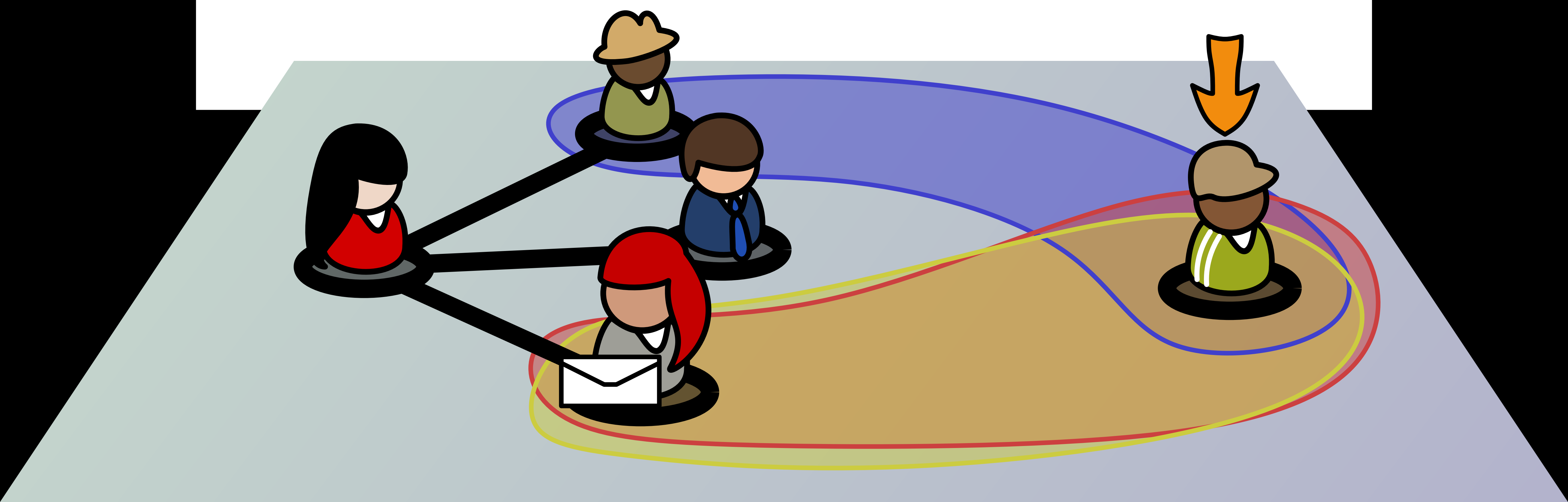
SIMPLE ROUTING

- Algorithm
  - Give message to neighbour who shares most categories with the target
  - $\max_{u:(s,u)\in E} |\{C : s \notin C \wedge u,t \in C\}|$



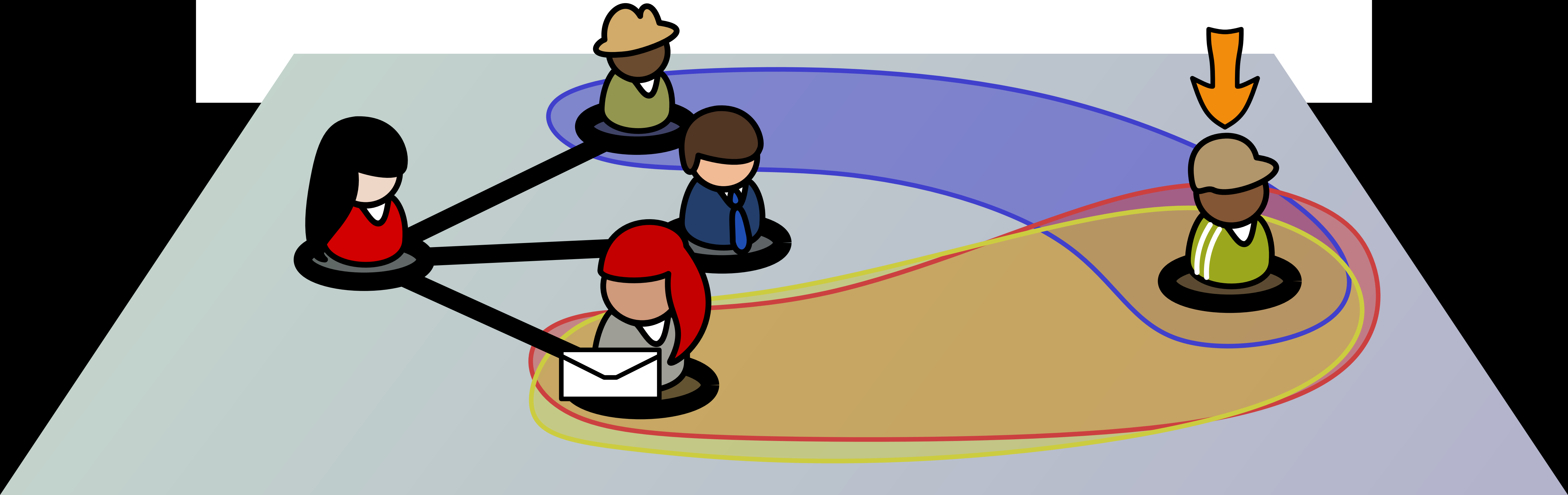
SIMPLE ROUTING

- Algorithm
  - Give message to neighbour who shares most categories with the target
  - $\max_{u:(s,u)\in E} |\{C : s \notin C \wedge u, t \in C\}|$



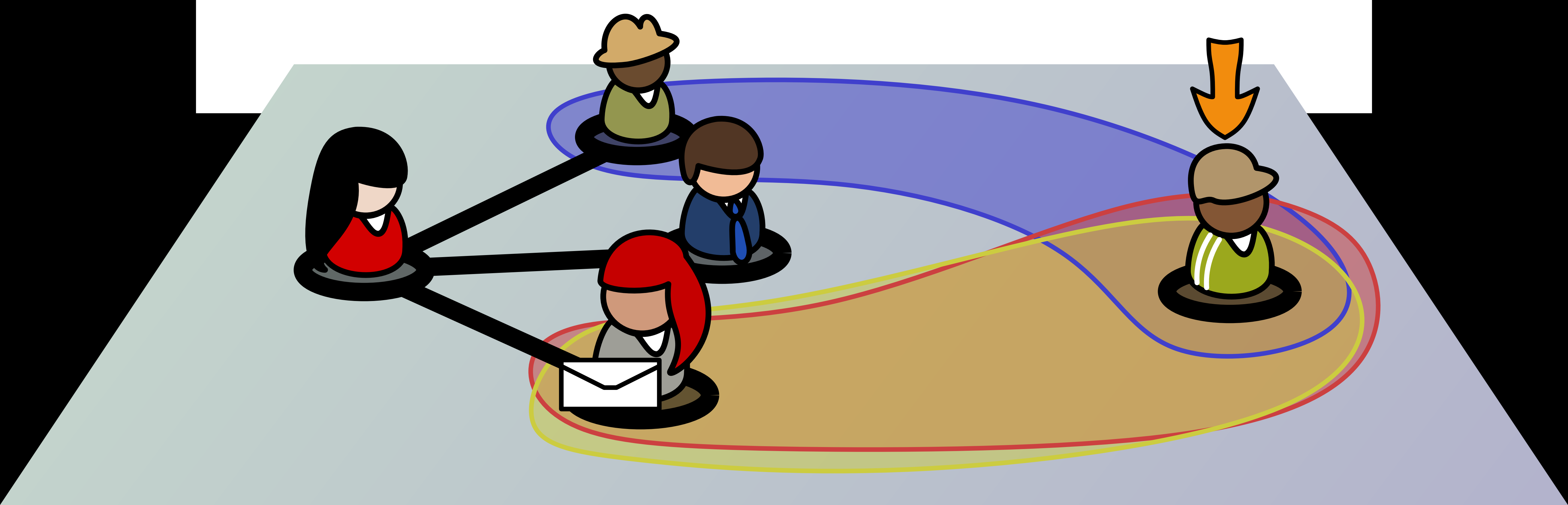
SIMPLE ROUTING

- Algorithm
  - Give message to neighbour who shares most categories with the target
  - $\max_{u:(s,u)\in E} |\{C : s \notin C \wedge u, t \in C\}|$
- Rationale



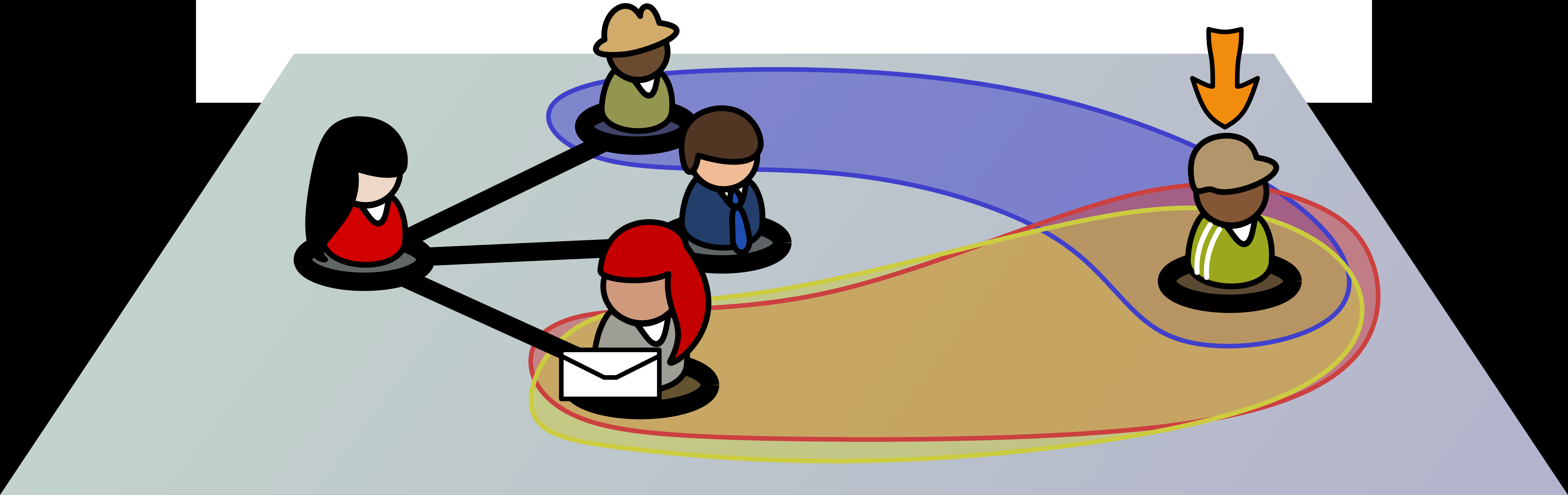
SIMPLE ROUTING

- Algorithm
  - Give message to neighbour who shares most categories with the target
  - $\max_{u:(s,u)\in E} |\{C : s \notin C \wedge u, t \in C\}|$
- Rationale
  - Simplest interpretation of "category-based" routing



SIMPLE ROUTING

- Algorithm
  - Give message to neighbour who shares most categories with the target
  - $\max_{u:(s,u)\in E} |\{C : s \notin C \wedge u,t \in C\}|$
- Rationale
  - Simplest interpretation of "category-based" routing
  - Requires only local knowledge about neighbours and target



INTERNALLY CONNECTED

INTERNALLY CONNECTED

- Definition

INTERNALLY CONNECTED

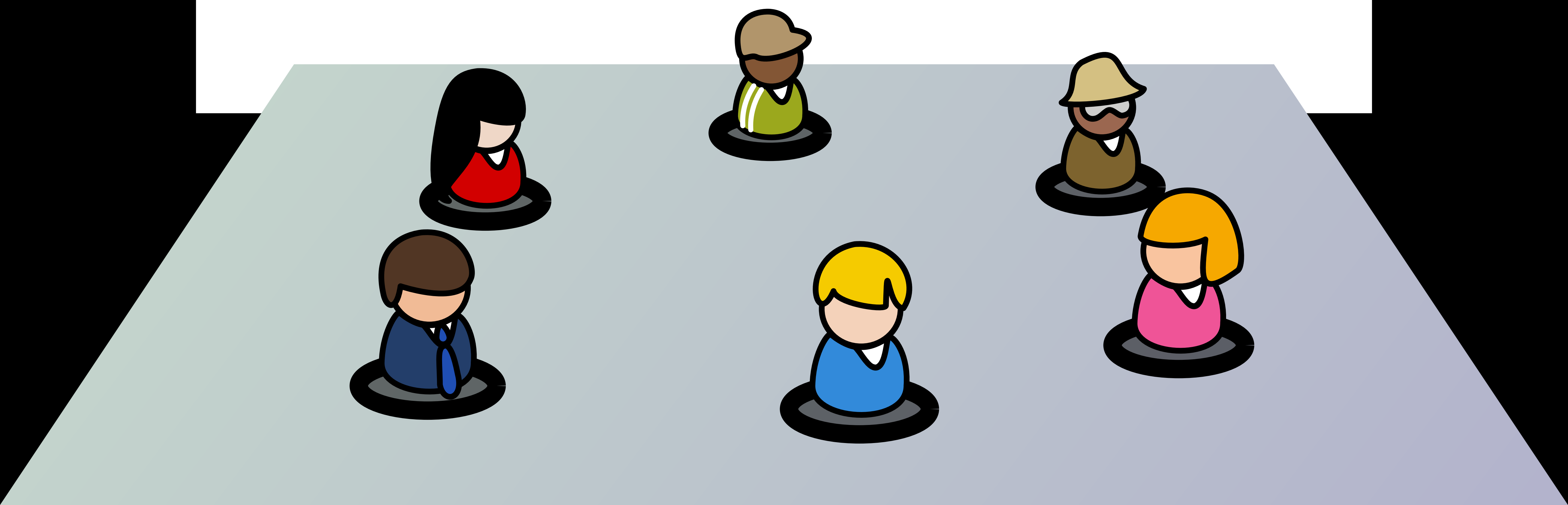
- Definition
  - The network is connected inside every category

INTERNALLY CONNECTED

- Definition
  - The network is connected inside every category
  - $\mathrm{inco}(G, \mathcal{S}) : \forall C \in \mathcal{S} : G[C]$ is connected

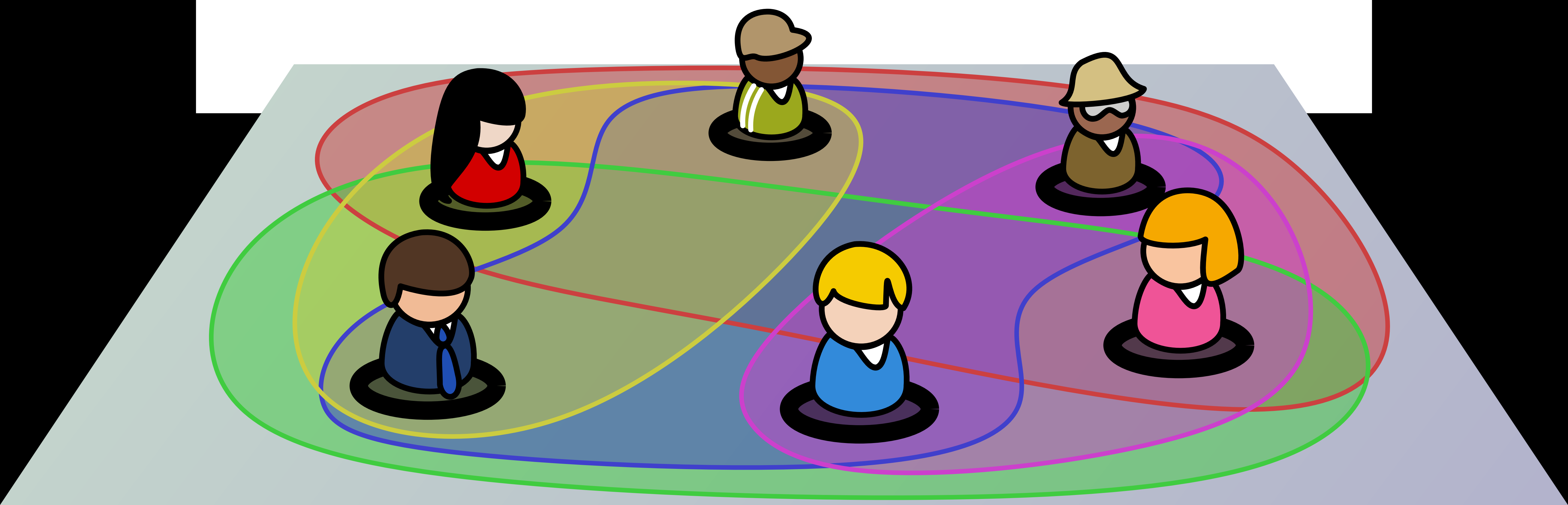
INTERNALLY CONNECTED

- Definition
  - The network is connected inside every category
  - $\mathrm{inco}(G, \mathcal{S})$: $\forall C \in \mathcal{S} : G[C]$ is connected



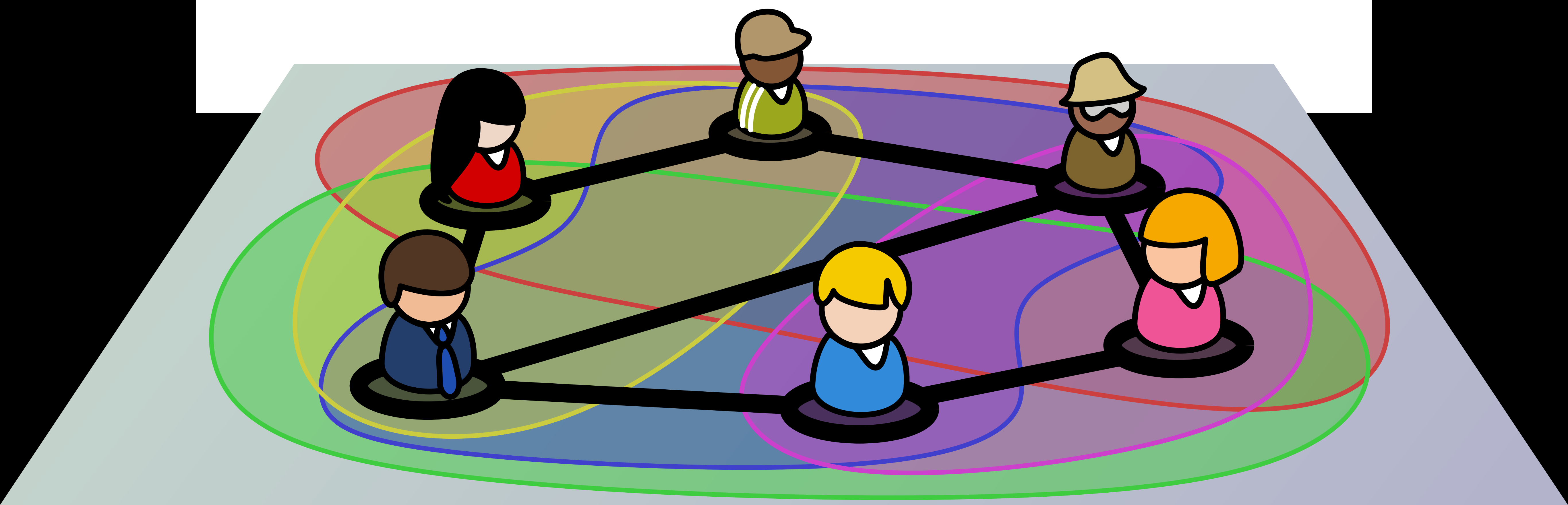
INTERNALLY CONNECTED

- Definition
  - The network is connected inside every category
  - $\mathrm{inco}(G, \mathcal{S})$: $\forall C \in \mathcal{S} : G[C]$ is connected



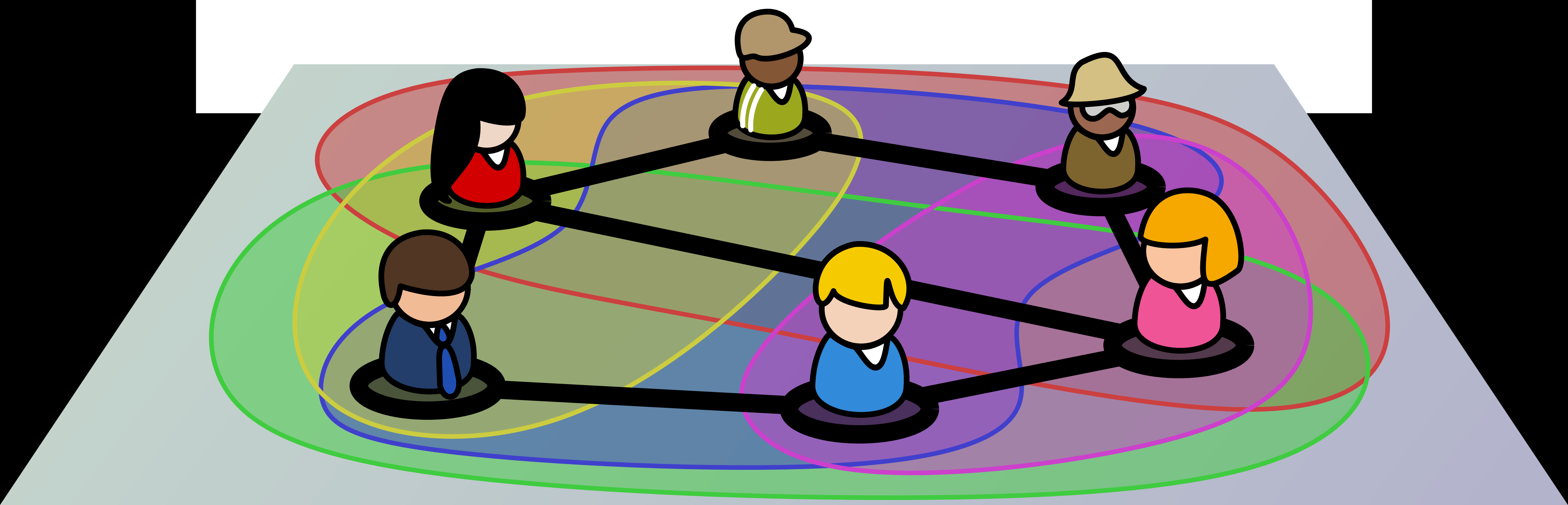
INTERNALLY CONNECTED

- Definition
  - The network is connected inside every category
  - $\mathrm{inco}(G, \mathcal{S})$: $\forall C \in \mathcal{S} : G[C]$ is connected



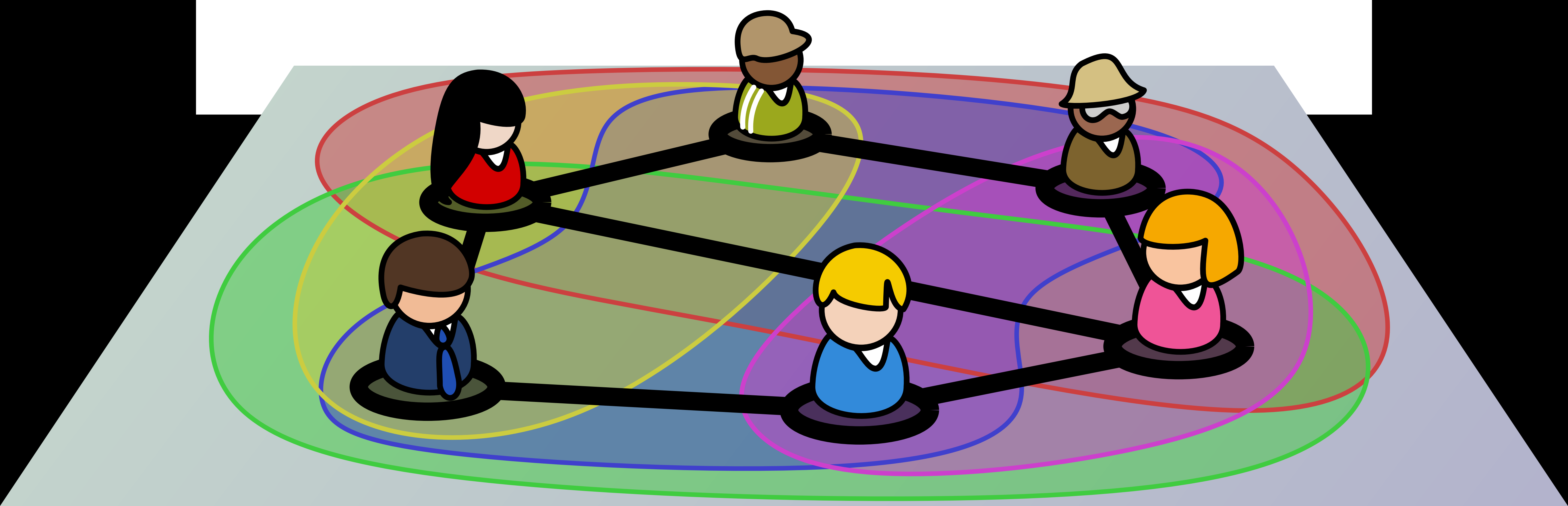
INTERNALLY CONNECTED

- Definition
  - The network is connected inside every category
  - $\mathrm{inco}(G, \mathcal{S})$: $\forall C \in \mathcal{S} : G[C]$ is connected



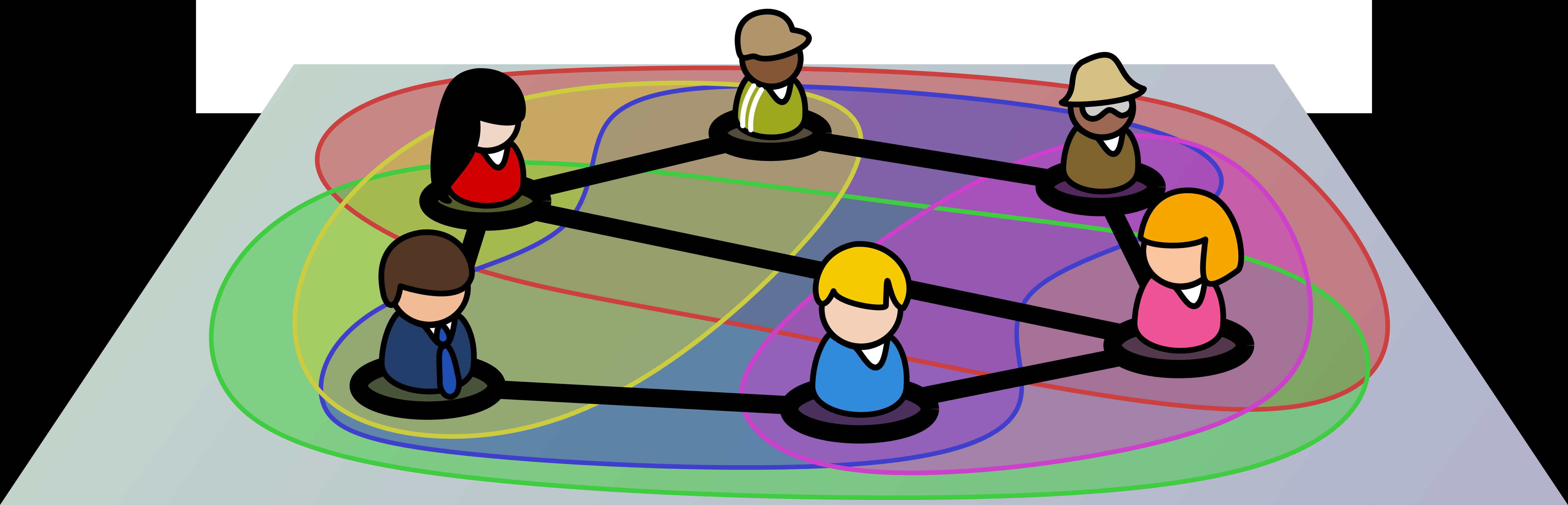
INTERNALLY CONNECTED

- Definition
  - The network is connected inside every category
  - $\mathrm{inco}(G, \mathcal{S})$: $\forall C \in \mathcal{S} : G[C]$ is connected
- Rationale



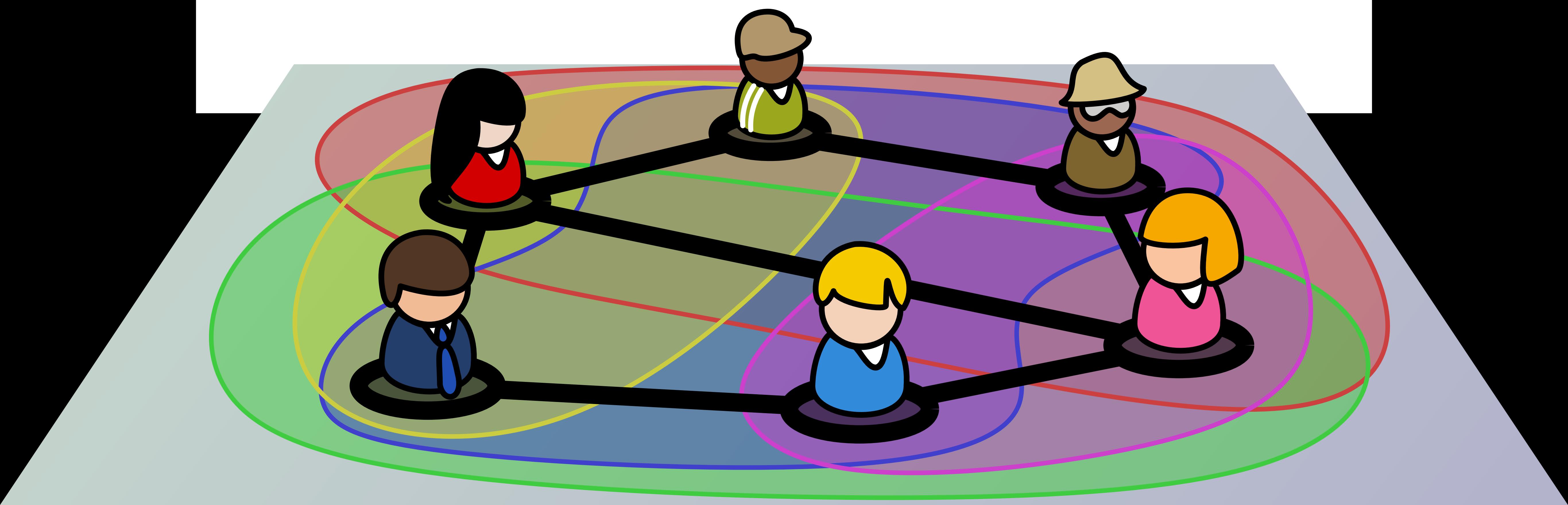
INTERNALLY CONNECTED

- Definition
  - The network is connected inside every category
  - $\text{inco}(G, \mathcal{S})$: $\forall C \in \mathcal{S} : G[C]$ is connected
- Rationale
  - Seems like a natural assumption

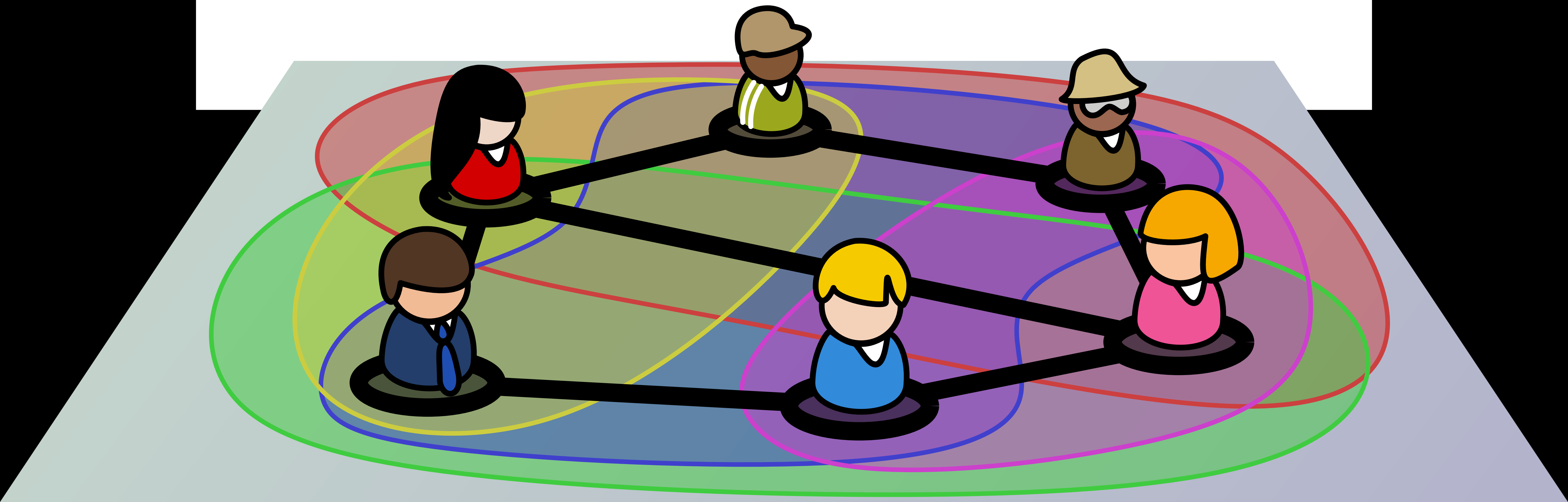


INTERNALLY CONNECTED

- Definition
  - The network is connected inside every category
  - $\mathrm{inco}(G, \mathcal{S})$: $\forall C \in \mathcal{S} : G[C]$ is connected
- Rationale
  - Seems like a natural assumption
  - Makes it a lot easier to reason about simple routing

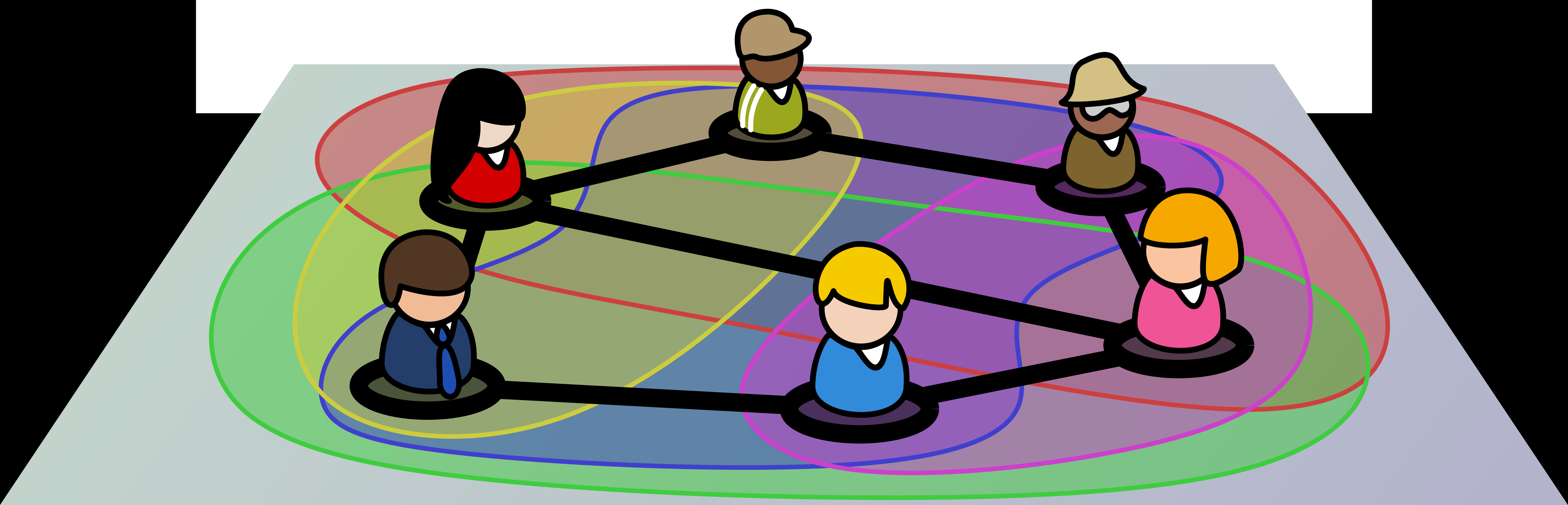


SHATTERED



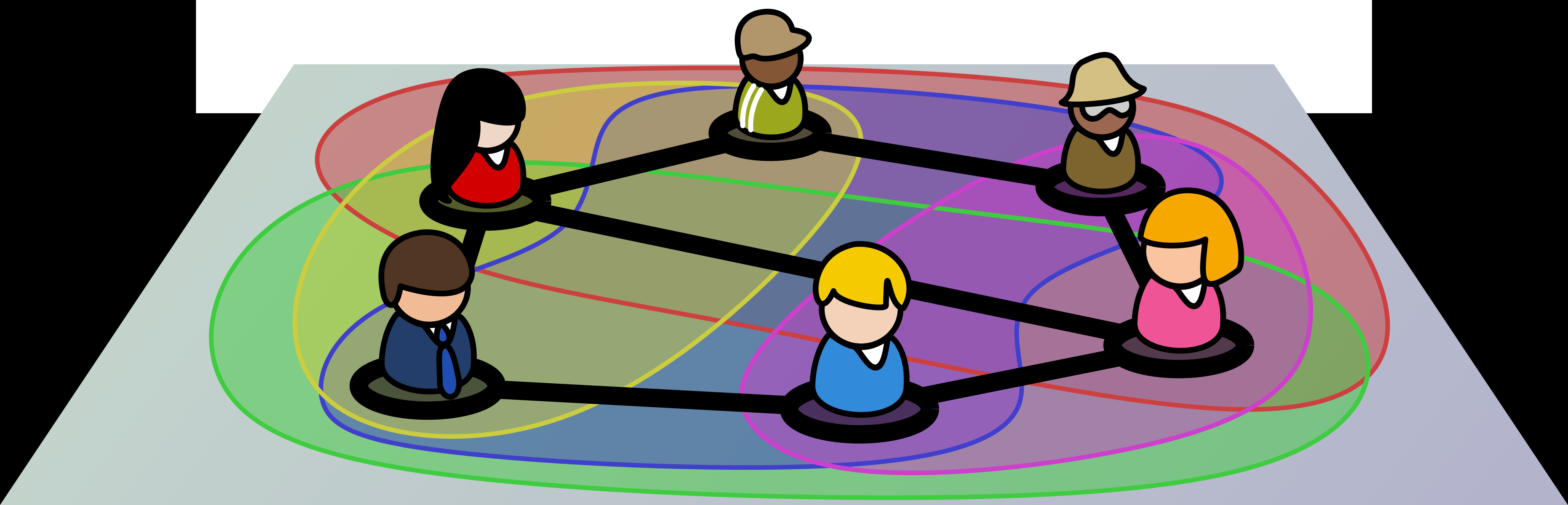
SHATTERED

- Definition



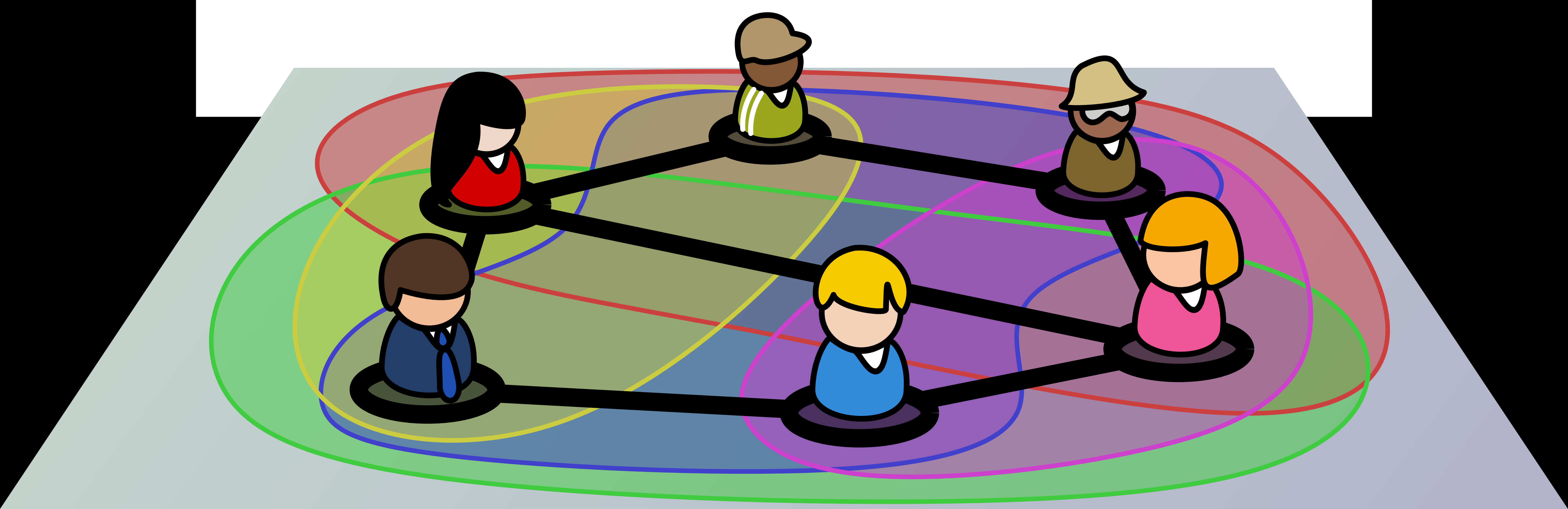
SHATTERED

- Definition
  - Everyone has a neighbour who shares a new category with everyone else



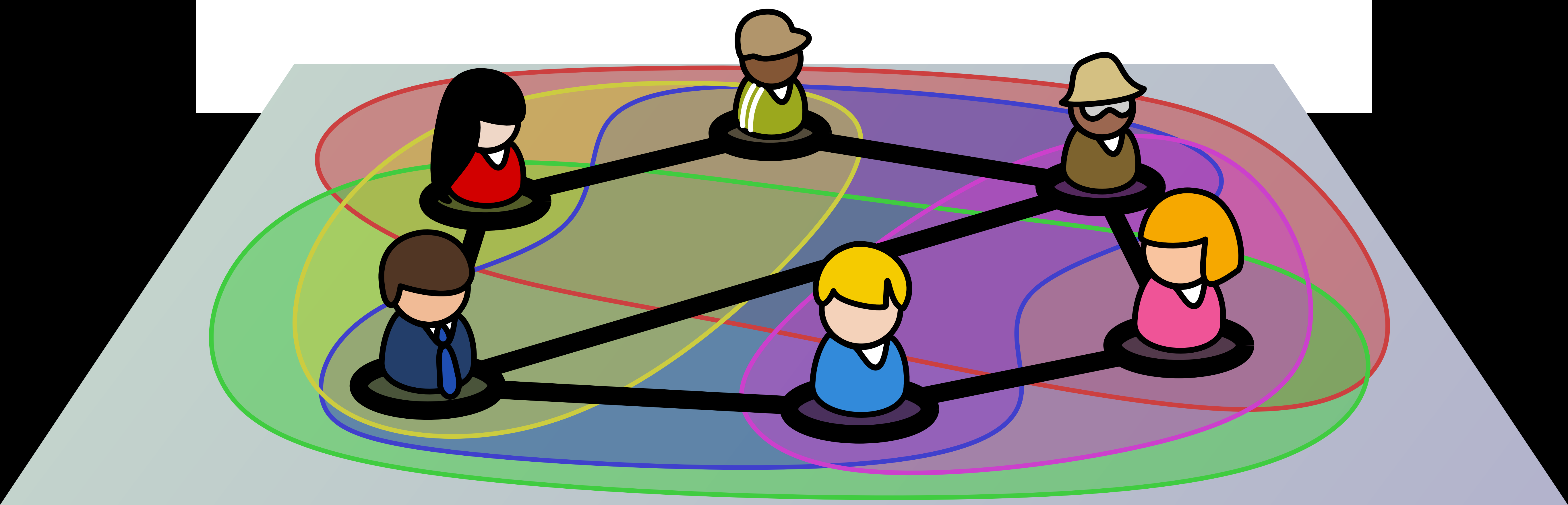
SHATTERED

- Definition
  - Everyone has a neighbour who shares a new category with everyone else
  - $\mathrm{shat}(G, \mathcal{S})$: $\forall s, t \exists u, C : (s, u) \in E \wedge s \notin C \wedge u, t \in C$



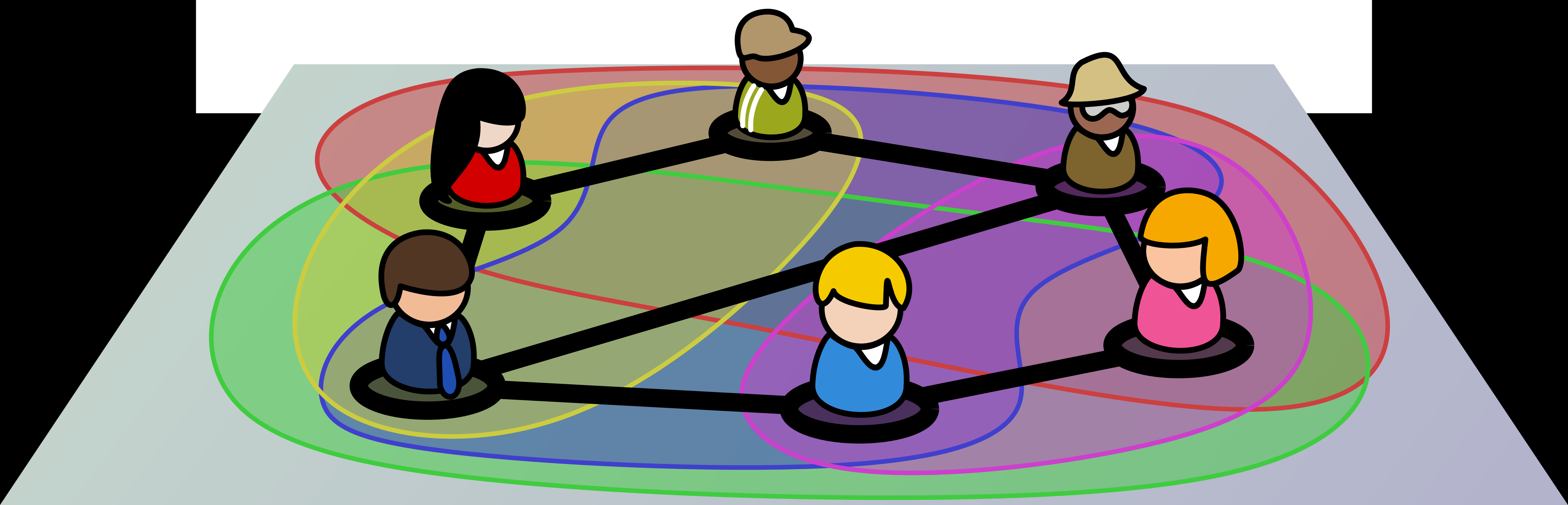
SHATTERED

- Definition
  - Everyone has a neighbour who shares a new category with everyone else
  - $\text{shat}(G, \mathcal{S})$: $\forall s, t \exists u, C : (s, u) \in E \wedge s \notin C \wedge u, t \in C$



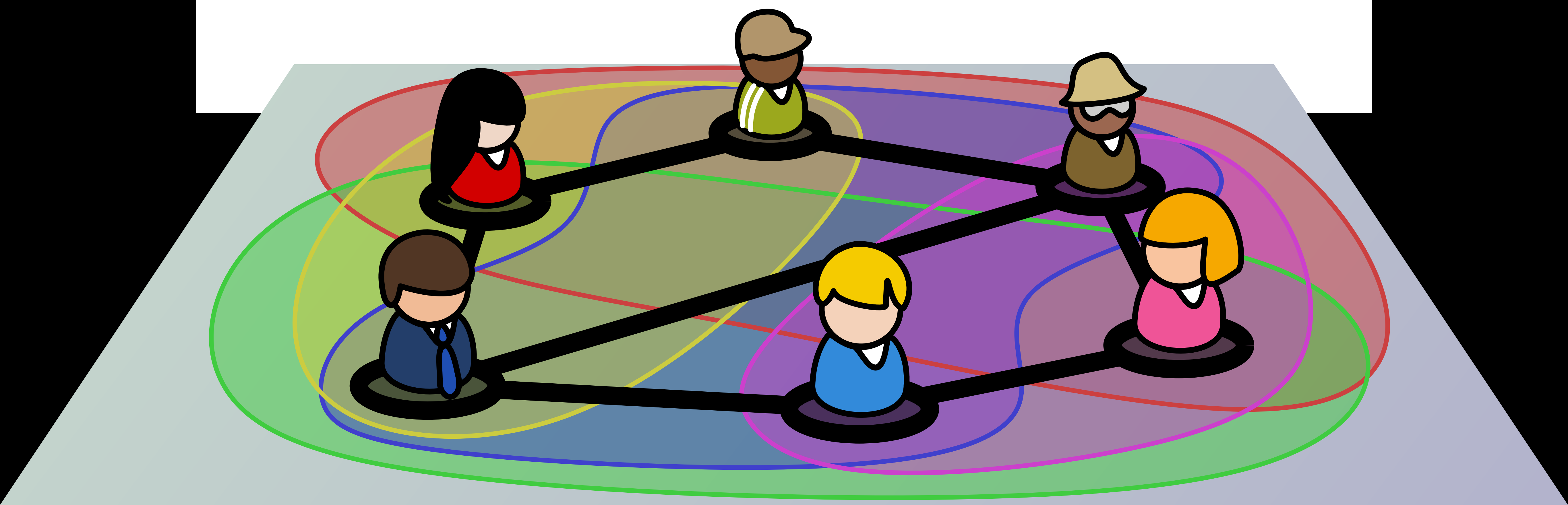
SHATTERED

- Definition
  - Everyone has a neighbour who shares a new category with everyone else
  - $\mathrm{shat}(G, \mathcal{S})$: $\forall s, t \exists u, C : (s, u) \in E \wedge s \notin C \wedge u, t \in C$
- Rationale



SHATTERED

- Definition
  - Everyone has a neighbour who shares a new category with everyone else
  - $\text{shat}(G, \mathcal{S})$: $\forall s, t \exists u, C : (s, u) \in E \wedge s \notin C \wedge u, t \in C$
- Rationale
  - Neccesary condition for routing to work



MEMBERSHIP DIMENSION

MEMBERSHIP DIMENSION

- Definition

MEMBERSHIP DIMENSION

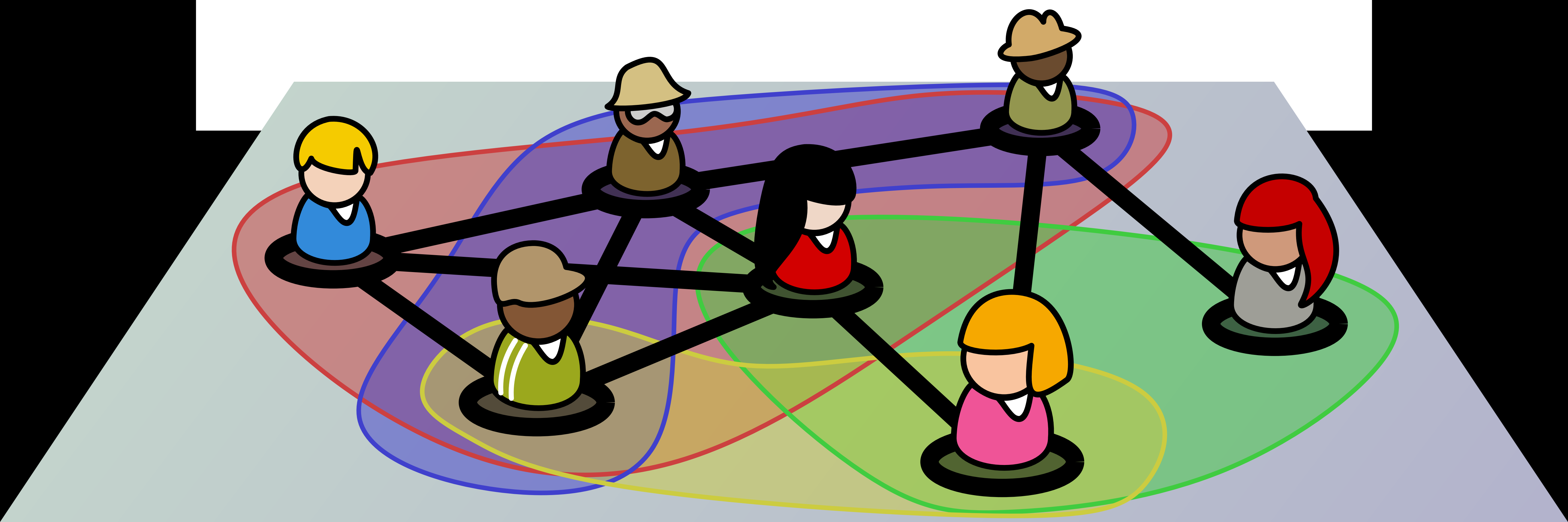
- Definition
  - Largest number of categories anyone is in

MEMBERSHIP DIMENSION

- Definition
  - Largest number of categories anyone is in
  - $\mathrm{mem}(\mathcal{S}) = \max_{u \in U} |\{C \in \mathcal{S} \mid u \in C\}|$

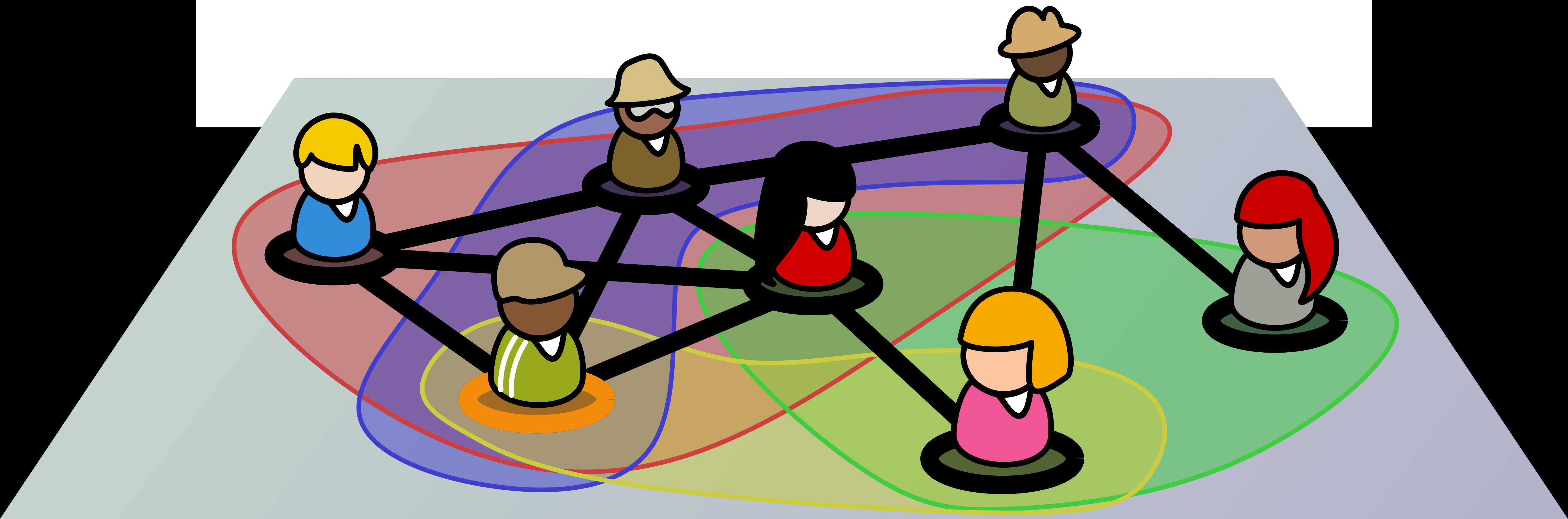
MEMBERSHIP DIMENSION

- Definition
  - Largest number of categories anyone is in
  - $\text{mem}(\mathcal{S}) = \max_{u \in U} |\{C \in \mathcal{S} \mid u \in C\}|$



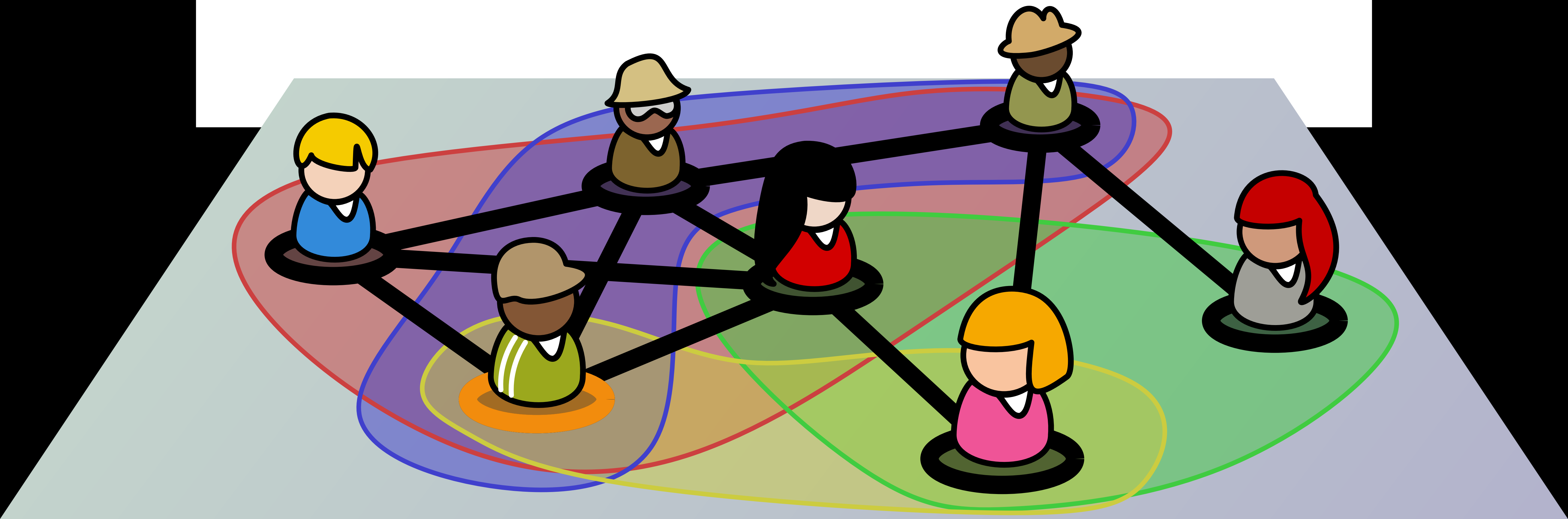
MEMBERSHIP DIMENSION

- Definition
  - Largest number of categories anyone is in
  - $\mathrm{mem}(\mathcal{S}) = \max_{u \in U} |\{C \in \mathcal{S} \mid u \in C\}|$



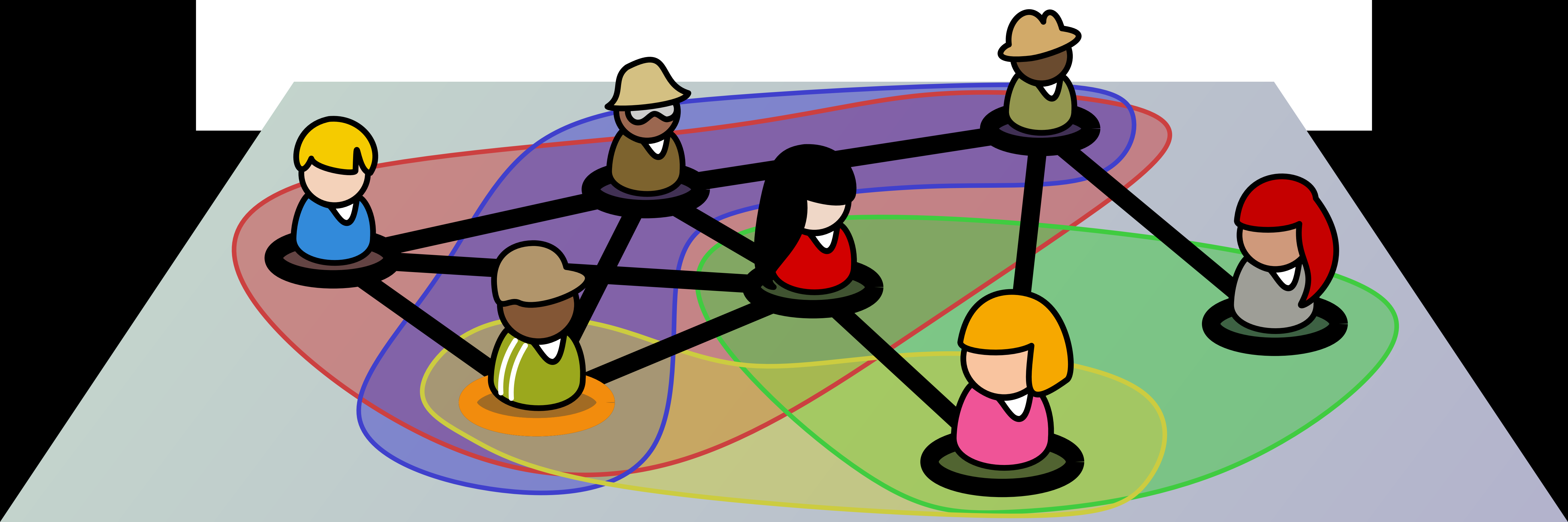
MEMBERSHIP DIMENSION

- Definition
  - Largest number of categories anyone is in
  - $\mathrm{mem}(\mathcal{S}) = \max_{u \in U} |\{C \in \mathcal{S} \mid u \in C\}|$
- Rationale



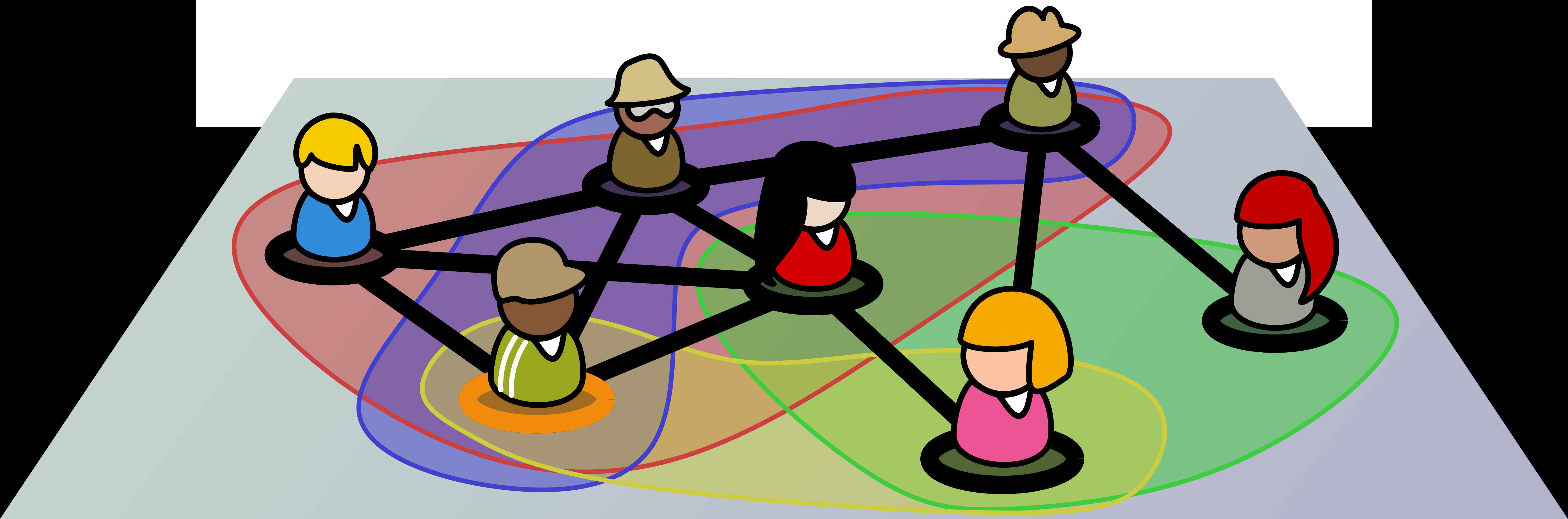
MEMBERSHIP DIMENSION

- Definition
  - Largest number of categories anyone is in
  - $\mathrm{mem}(\mathcal{S}) = \max_{u \in U} |\{C \in \mathcal{S} \mid u \in C\}|$
- Rationale
  - Captures the "cognitive load" of people



MEMBERSHIP DIMENSION

- Definition
  - Largest number of categories anyone is in
  - $\mathrm{mem}(\mathcal{S}) = \max_{u \in U} |\{C \in \mathcal{S} \mid u \in C\}|$
- Rationale
  - Captures the "cognitive load" of people
  - We expect the membership dimension to be *small*



RESULTS

RESULTS

- Simple routing works?

RESULTS

- Simple routing works?
  - yes $\rightarrow \mathrm{shat}(G, \mathcal{S})$

RESULTS

- Simple routing works?
  - yes $\rightarrow$ $\mathrm{shat}(G, \mathcal{S})$
  - $\mathrm{shat}(G, \mathcal{S}) \wedge \exists T : \mathrm{inco}(T, \mathcal{S}) \rightarrow$ yes

RESULTS

- Simple routing works?
  - $\text{yes} \rightarrow \text{shat}(G, \mathcal{S})$
  - $\text{shat}(G, \mathcal{S}) \wedge \exists T : \text{inco}(T, \mathcal{S}) \rightarrow \text{yes}$
- Bounds on membership dimension

RESULTS

- Simple routing works?
  - $\text{yes} \rightarrow \text{shat}(G, \mathcal{S})$
  - $\text{shat}(G, \mathcal{S}) \wedge \exists T : \text{inco}(T, \mathcal{S}) \rightarrow \text{yes}$
- Bounds on membership dimension
  - $\exists G \exists \mathcal{S} : \text{yes} \wedge \text{mem}(\mathcal{S}) = 1$

RESULTS

- Simple routing works?
  - yes $\rightarrow \mathrm{shat}(G, \mathcal{S})$
  - $\mathrm{shat}(G, \mathcal{S}) \wedge \exists T : \mathrm{inco}(T, \mathcal{S}) \rightarrow$ yes
- Bounds on membership dimension
  - $\exists G \exists \mathcal{S}$: yes $\wedge\, \mathrm{mem}(\mathcal{S}) = 1$
  - $\forall G \forall \mathcal{S}$: yes $\rightarrow \mathrm{mem}(\mathcal{S}) \geq \mathrm{diam}(G)$

RESULTS

- Simple routing works?
  - $\text{yes} \rightarrow \text{shat}(G, \mathcal{S})$
  - $\text{shat}(G, \mathcal{S}) \wedge \exists T : \text{inco}(T, \mathcal{S}) \rightarrow \text{yes}$
- Bounds on membership dimension
  - $\exists G \exists \mathcal{S} : \text{yes} \wedge \text{mem}(\mathcal{S}) = 1$
  - $\forall G \forall \mathcal{S} : \text{yes} \rightarrow \text{mem}(\mathcal{S}) \geq \text{diam}(G)$
  - $\forall G \exists \mathcal{S} : \text{yes} \wedge \text{mem}(\mathcal{S}) \leq (\text{diam}(G) + \log n)^2$

RESULTS

- Simple routing works?
  - $\text{yes} \to \text{shat}(G, \mathcal{S})$
  - $\text{shat}(G, \mathcal{S}) \wedge \exists T : \text{inco}(T, \mathcal{S}) \to \text{yes}$
- Bounds on membership dimension
  - $\exists G \exists \mathcal{S} : \text{yes} \wedge \text{mem}(\mathcal{S}) = 1$
  - $\forall G \forall \mathcal{S} : \text{yes} \to \text{mem}(\mathcal{S}) \geq \text{diam}(G)$
  - $\forall G \exists \mathcal{S} : \text{yes} \wedge \text{mem}(\mathcal{S}) \leq (\text{diam}(G) + \log n)^2$
  - $\forall \mathcal{S} \exists G : \text{no}$

PART III
TECHNICAL DETAILS

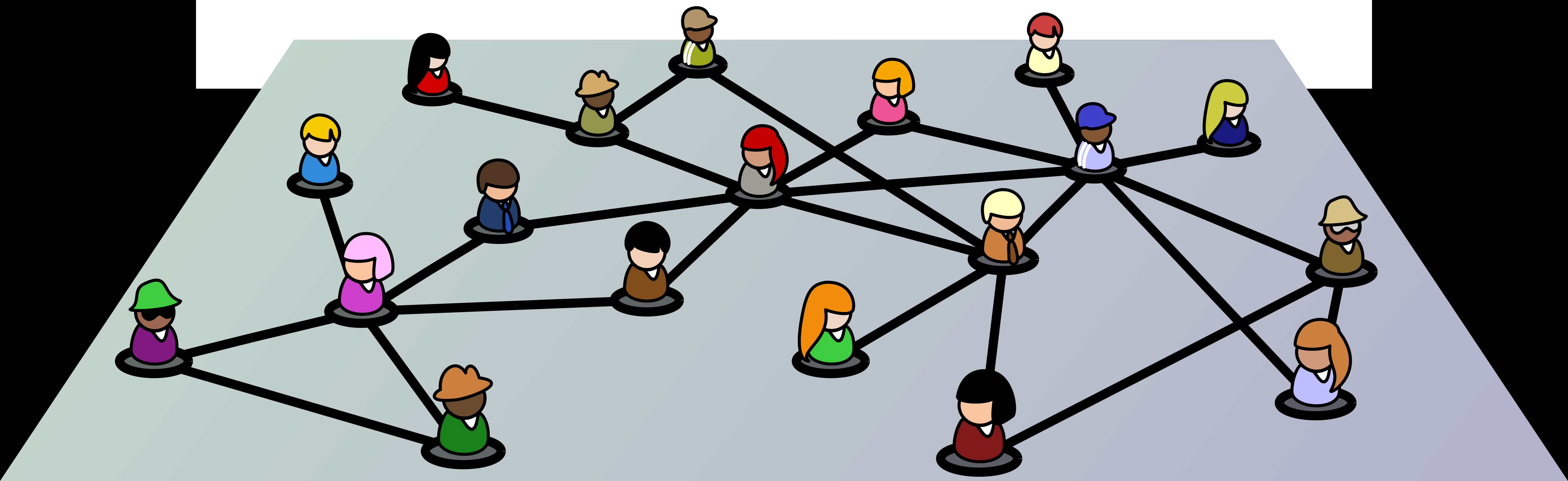
CATEGORY CONSTRUCTION

CATEGORY CONSTRUCTION

- Start with arbitrary graph G

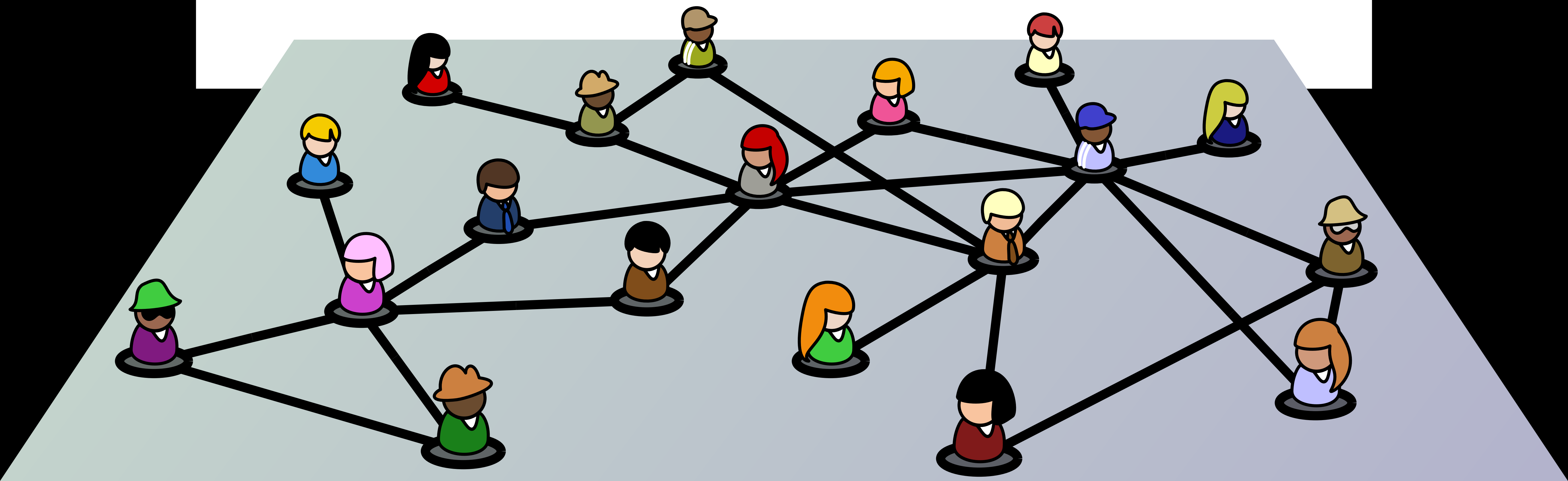
CATEGORY CONSTRUCTION

- Start with arbitrary graph G



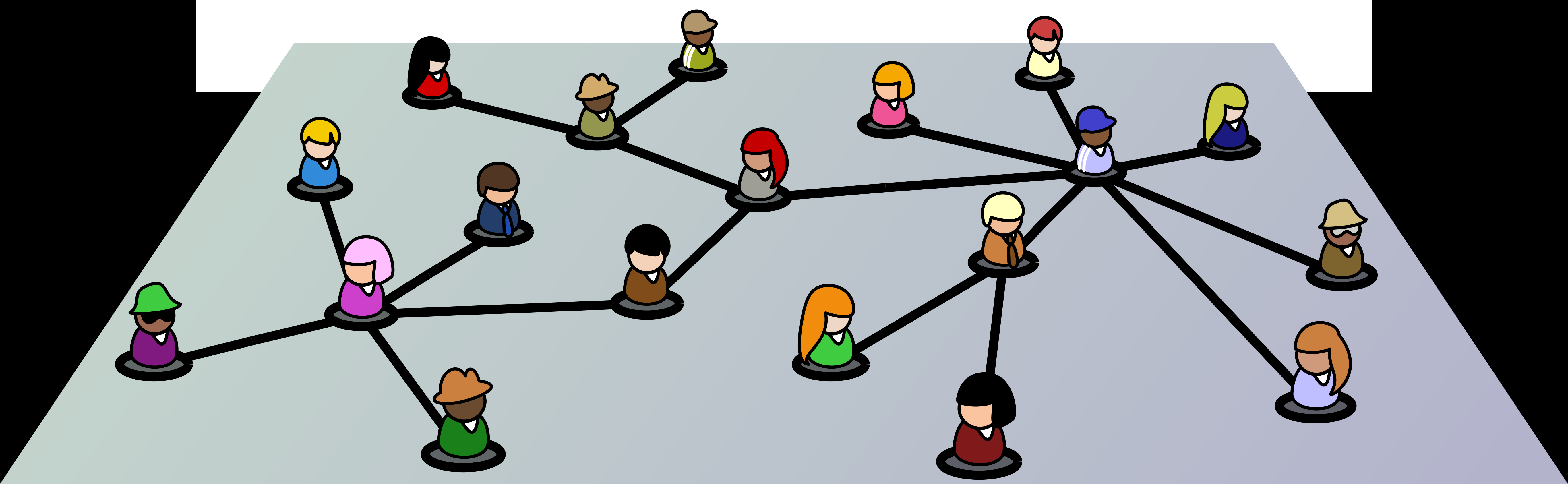
CATEGORY CONSTRUCTION

- Start with arbitrary graph G
- Compute spanning tree T of G



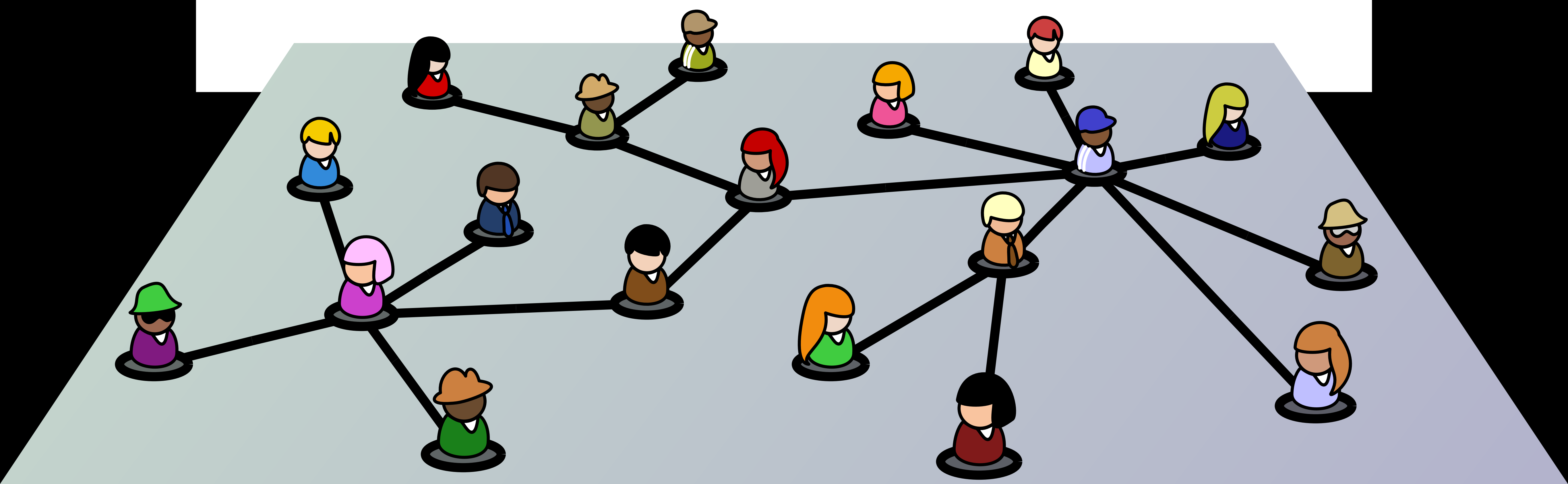
CATEGORY CONSTRUCTION

- Start with arbitrary graph G
- Compute spanning tree T of G



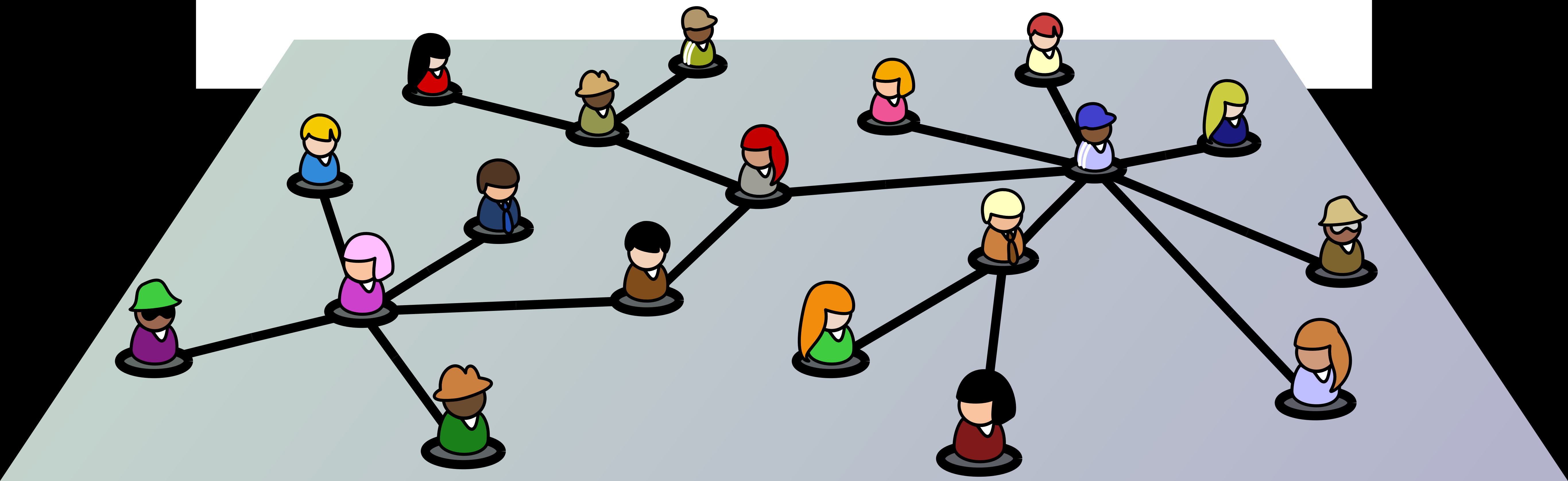
CATEGORY CONSTRUCTION

- Start with arbitrary graph G
- Compute spanning tree T of G
  - $\text{diam}(T) \leq 2\text{diam}(G)$



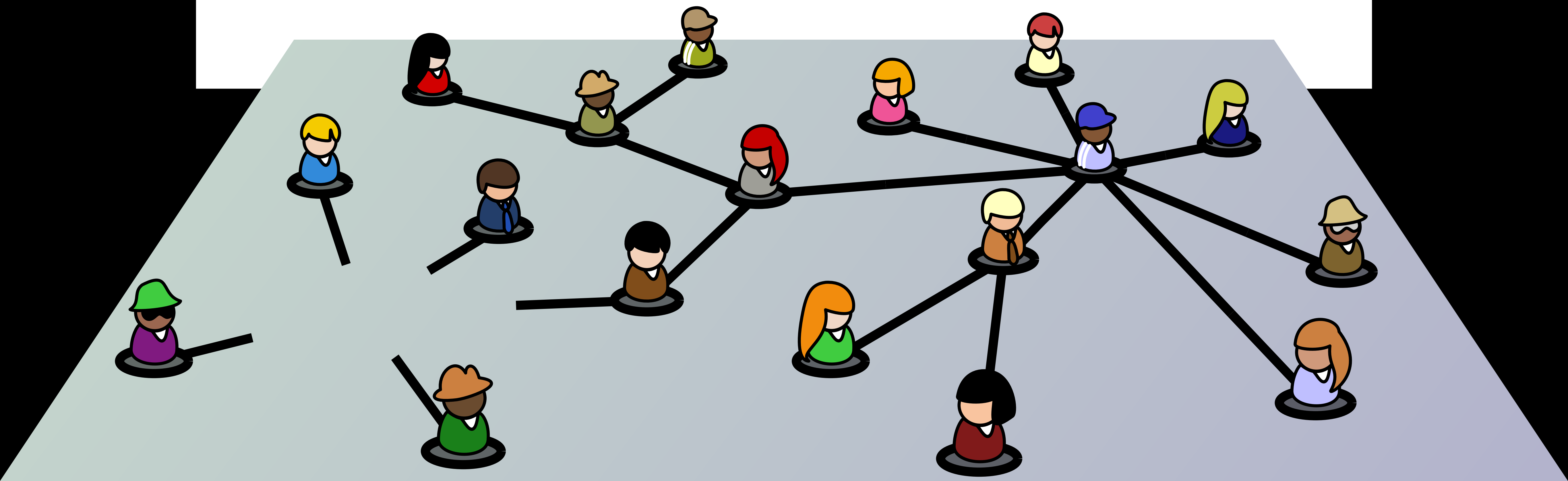
CATEGORY CONSTRUCTION

- Start with arbitrary graph G
- Compute spanning tree T of G
  - $\mathrm{diam}(T) \leq 2\mathrm{diam}(G)$
- Embed T in binary tree B



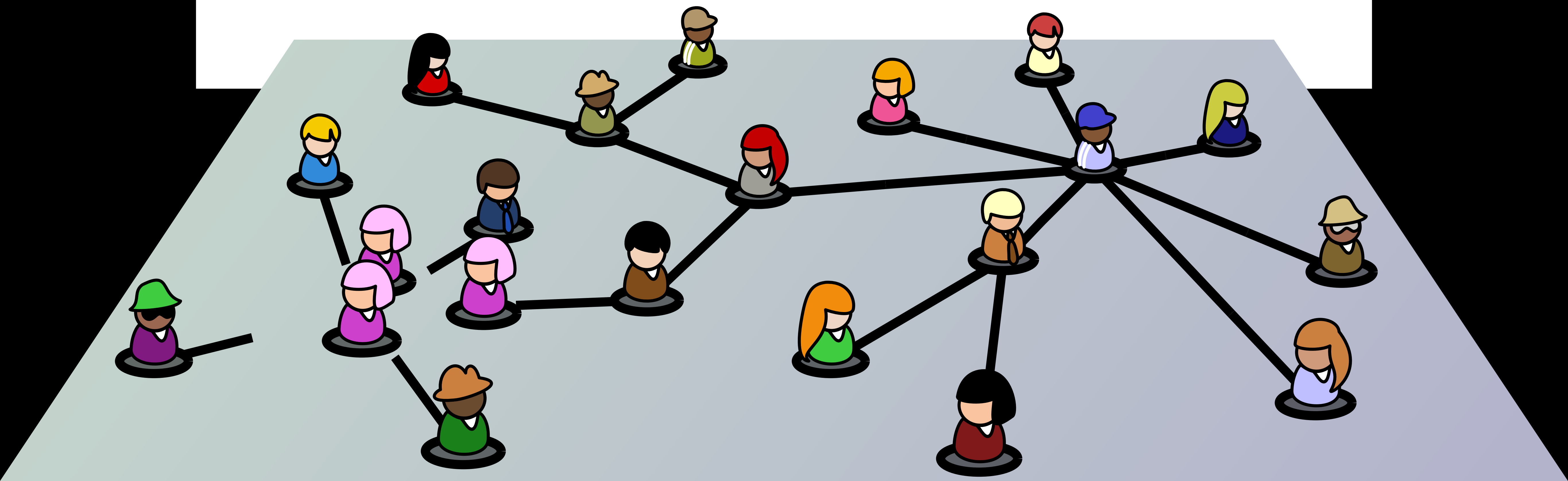
CATEGORY CONSTRUCTION

- Start with arbitrary graph G
- Compute spanning tree T of G
  - $\mathrm{diam}(T) \leq 2\mathrm{diam}(G)$
- Embed T in binary tree B



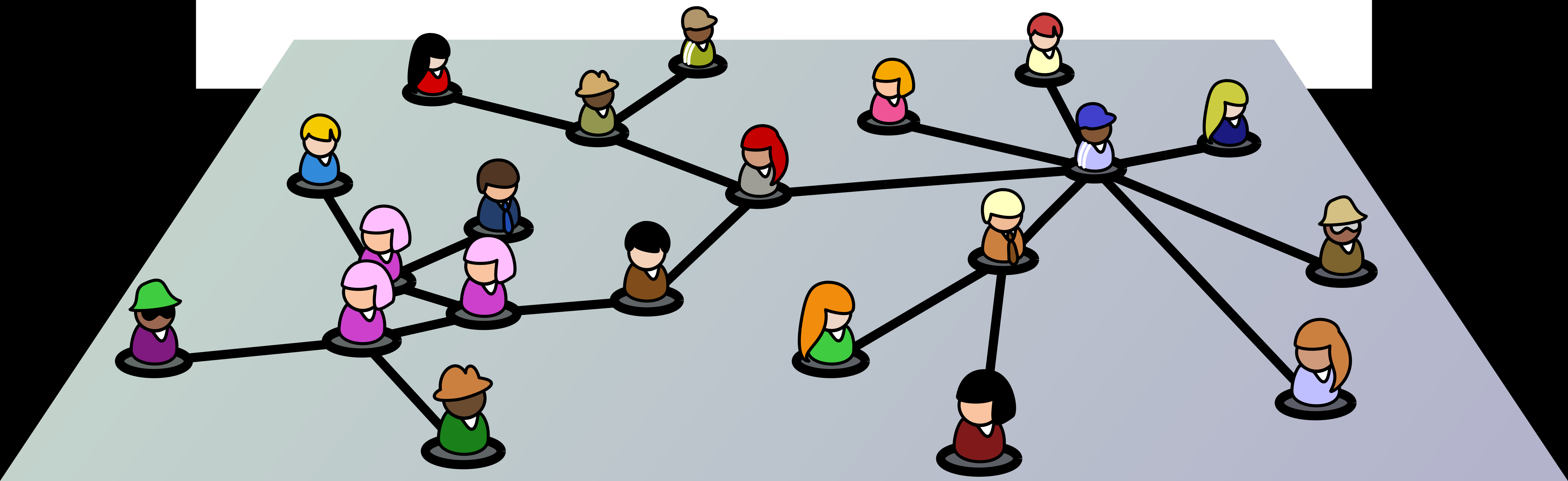
CATEGORY CONSTRUCTION

- Start with arbitrary graph G
- Compute spanning tree T of G
  - $\mathrm{diam}(T) \leq 2\mathrm{diam}(G)$
- Embed T in binary tree B



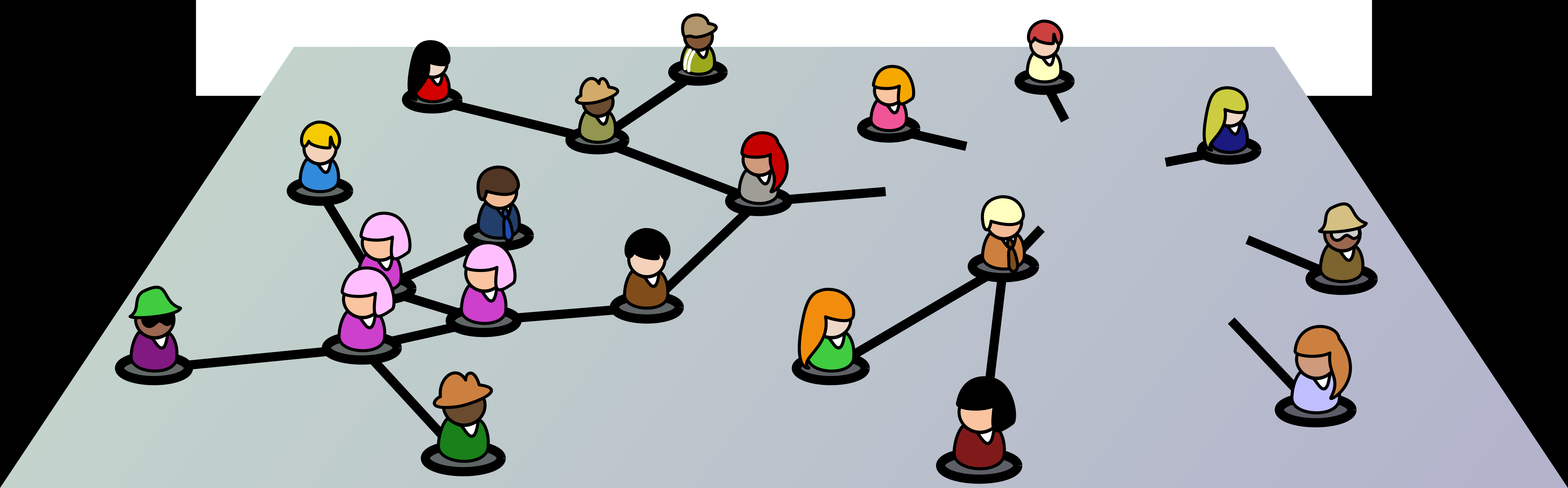
CATEGORY CONSTRUCTION

- Start with arbitrary graph G
- Compute spanning tree T of G
  - $\text{diam}(T) \leq 2\text{diam}(G)$
- Embed T in binary tree B



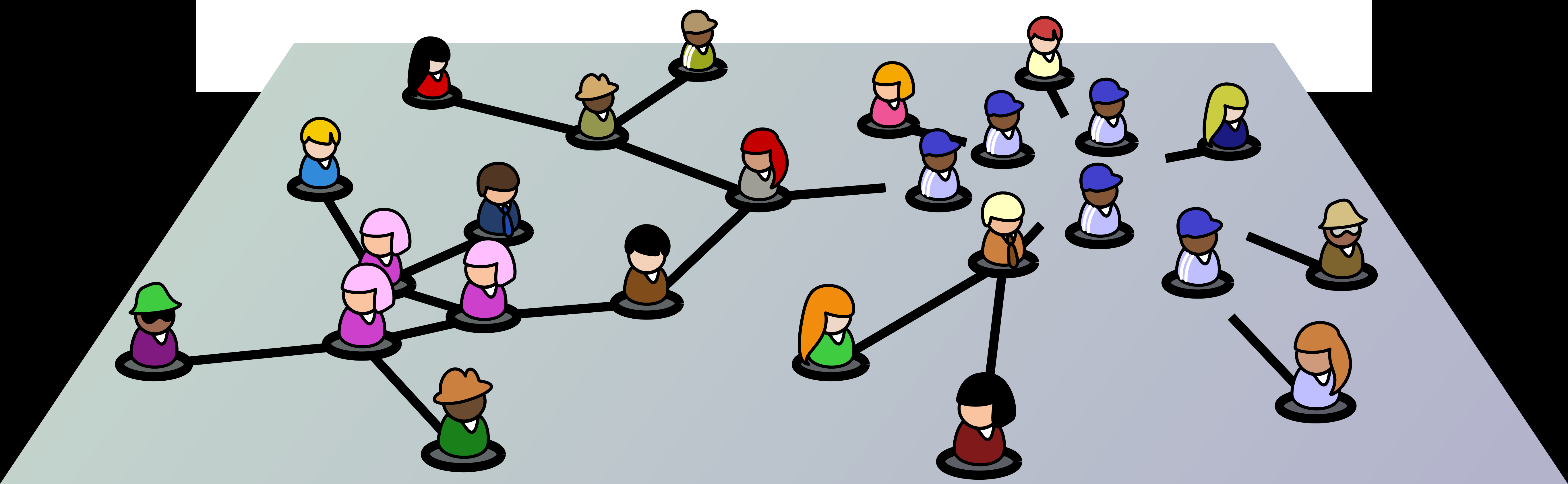
CATEGORY CONSTRUCTION

- Start with arbitrary graph G
- Compute spanning tree T of G
  - $\mathrm{diam}(T) \leq 2\mathrm{diam}(G)$
- Embed T in binary tree B



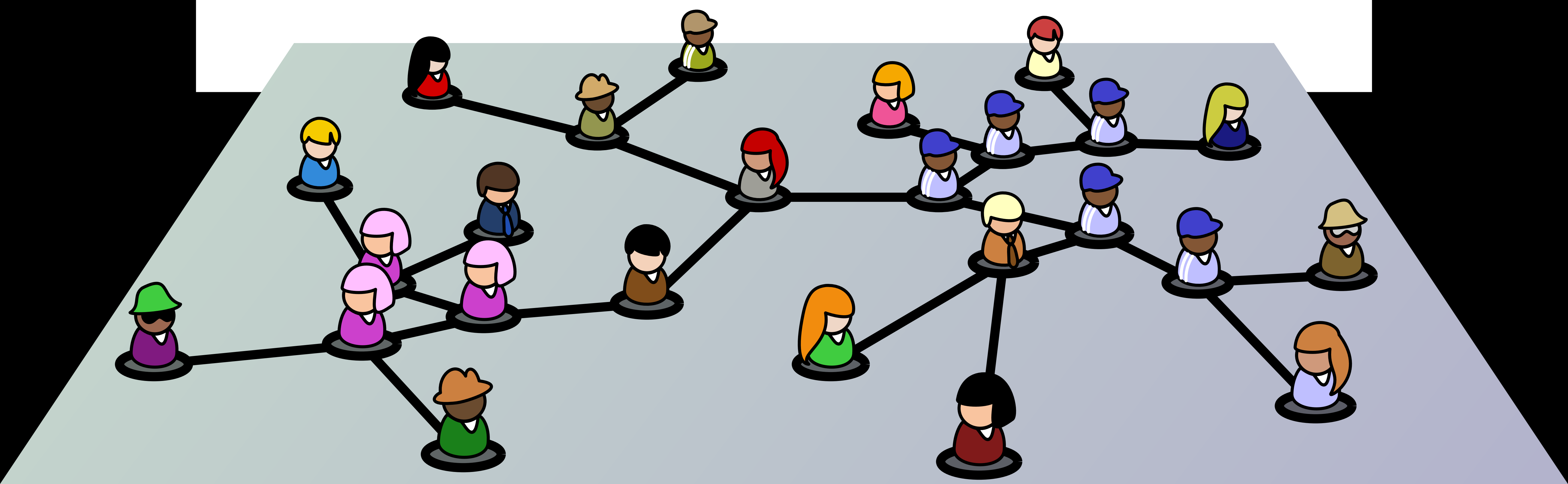
CATEGORY CONSTRUCTION

- Start with arbitrary graph G
- Compute spanning tree T of G
  - $\mathrm{diam}(T) \leq 2\mathrm{diam}(G)$
- Embed T in binary tree B



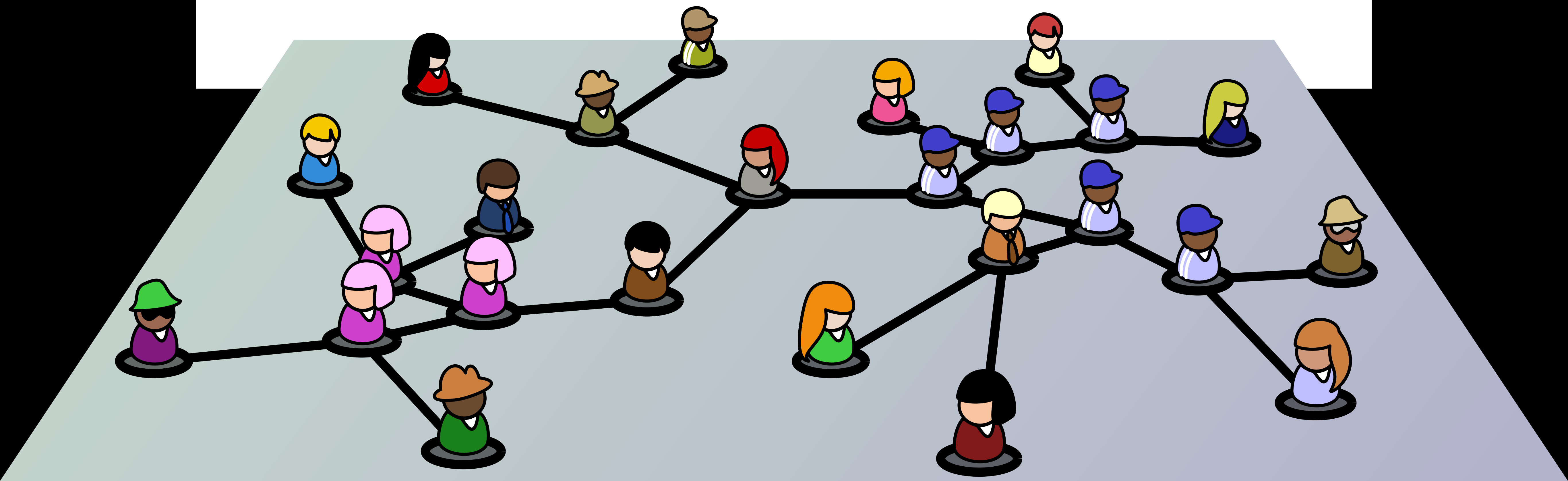
CATEGORY CONSTRUCTION

- Start with arbitrary graph G
- Compute spanning tree T of G
  - $\mathrm{diam}(T) \leq 2\mathrm{diam}(G)$
- Embed T in binary tree B

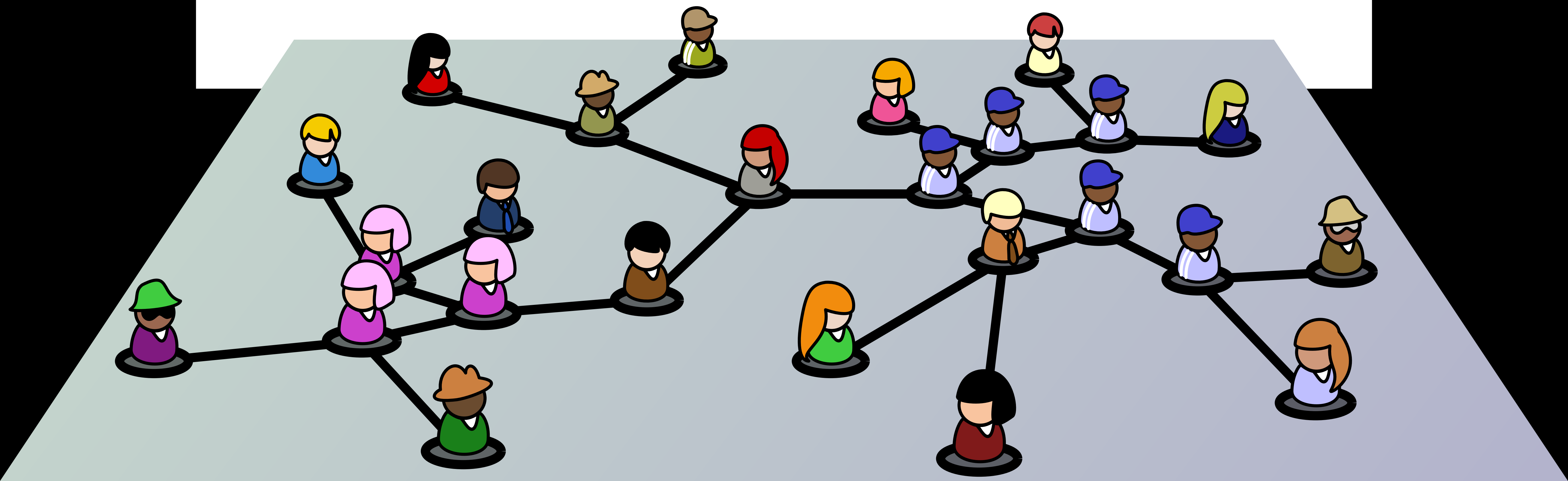


CATEGORY CONSTRUCTION

- Start with arbitrary graph G
- Compute spanning tree T of G
  - $\mathrm{diam}(T) \leq 2\mathrm{diam}(G)$
- Embed T in binary tree B
  - $\mathrm{diam}(B) \leq \mathrm{diam}(T) + \log n$

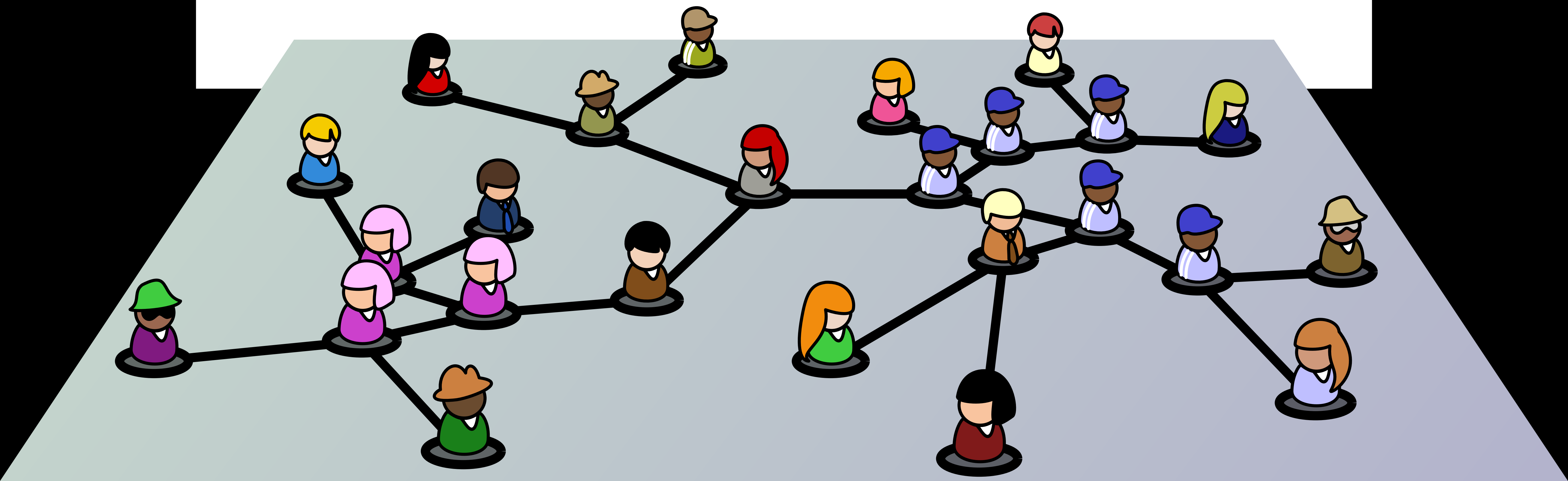


CATEGORY CONSTRUCTION



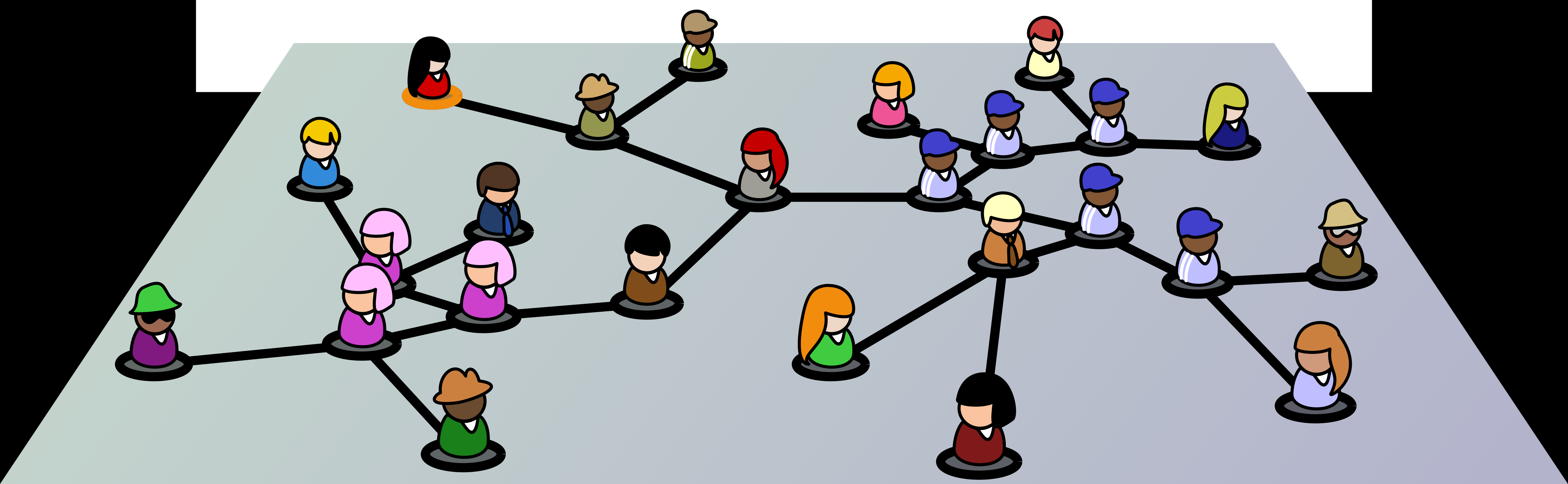
CATEGORY CONSTRUCTION

- Choose an arbitrary root for B



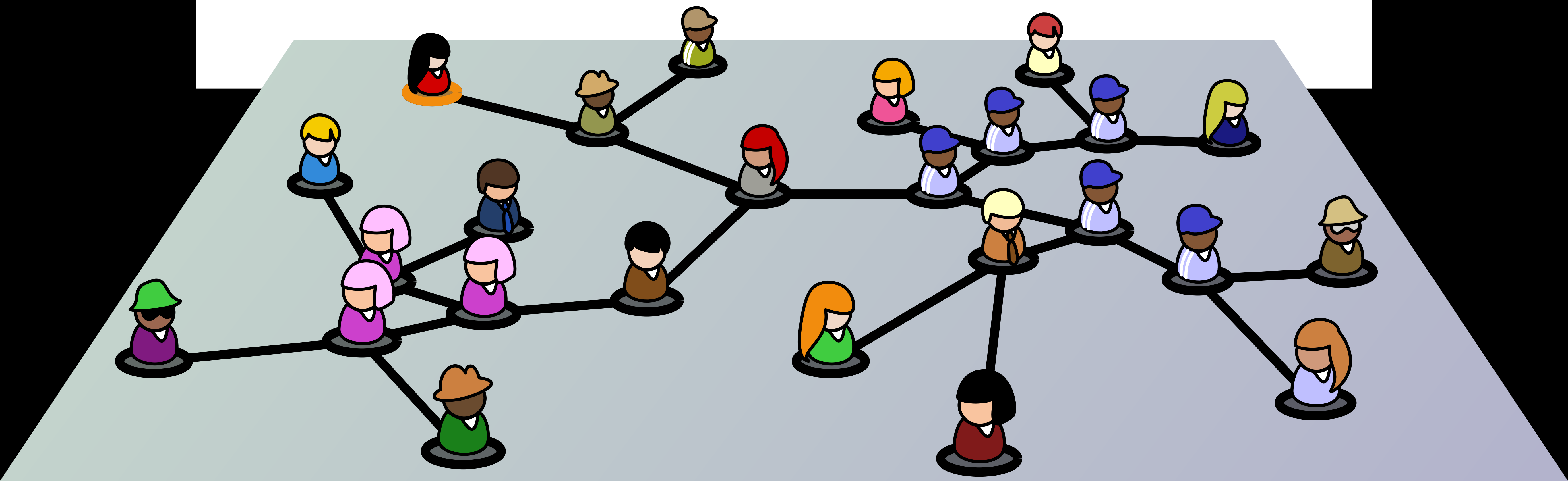
CATEGORY CONSTRUCTION

- Choose an arbitrary root for B



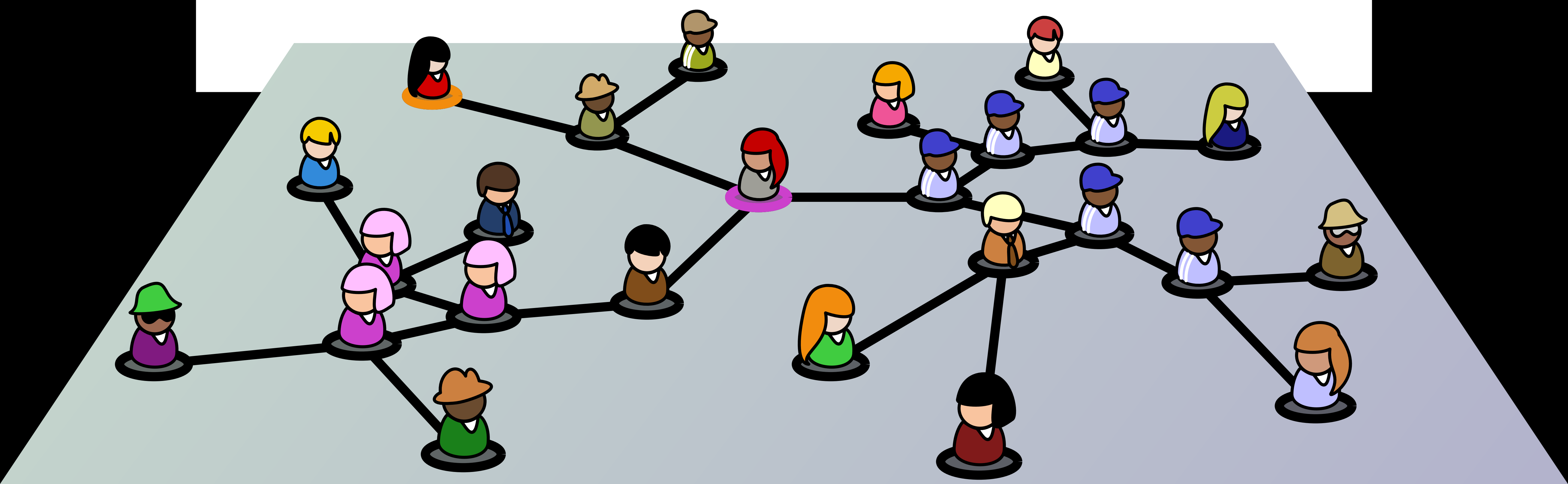
CATEGORY CONSTRUCTION

- Choose an arbitrary root for B
- For every node v in B and integer $d \leq \mathrm{diam}(B)$ create two categories



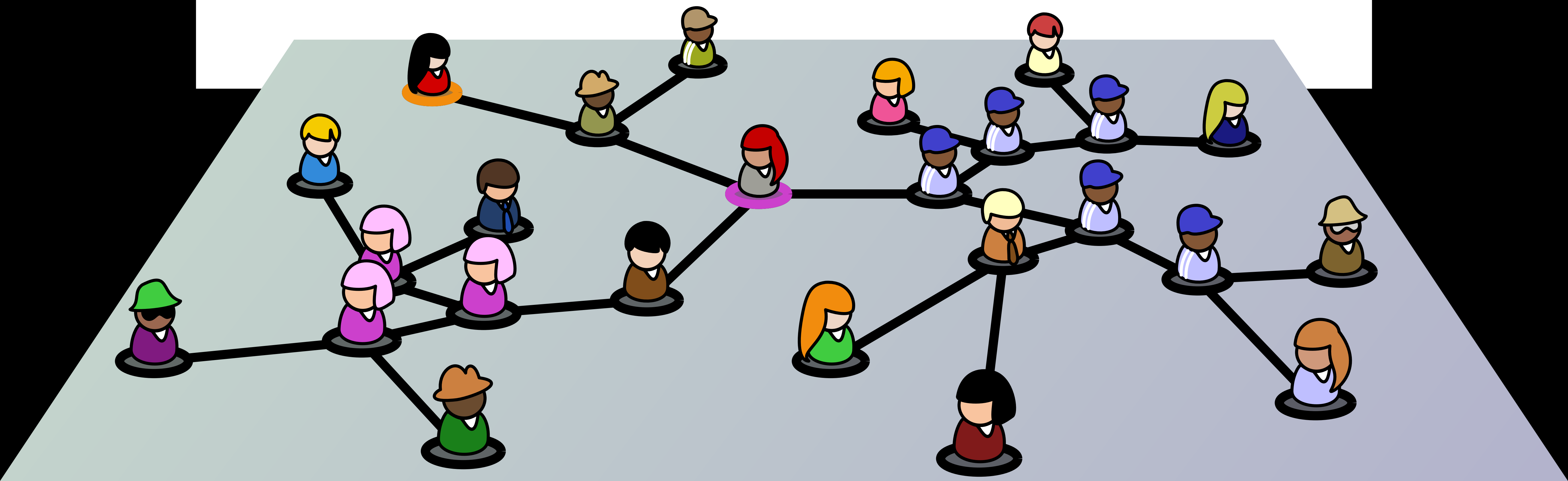
CATEGORY CONSTRUCTION

- Choose an arbitrary root for B
- For every node v in B and integer $d \leq \mathrm{diam}(B)$ create two categories



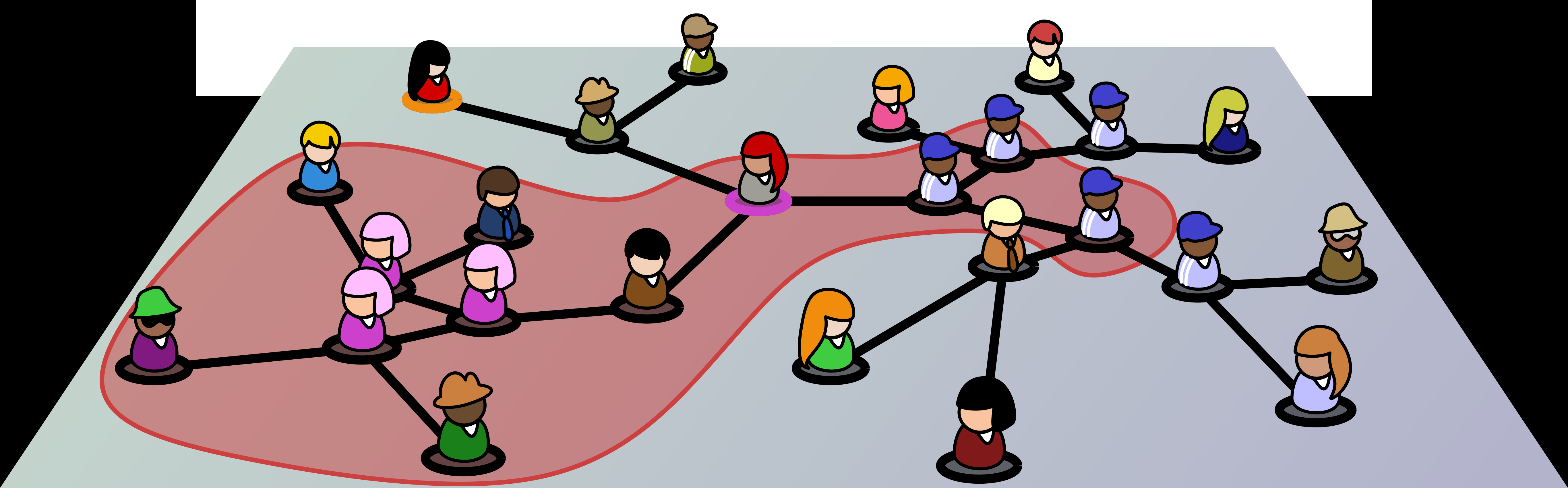
CATEGORY CONSTRUCTION

- Choose an arbitrary root for B
- For every node v in B and integer $d \leq \mathrm{diam}(B)$ create two categories
  - One contains all nodes in the left subtree of B at v and the first d levels of the right subtree of B at v



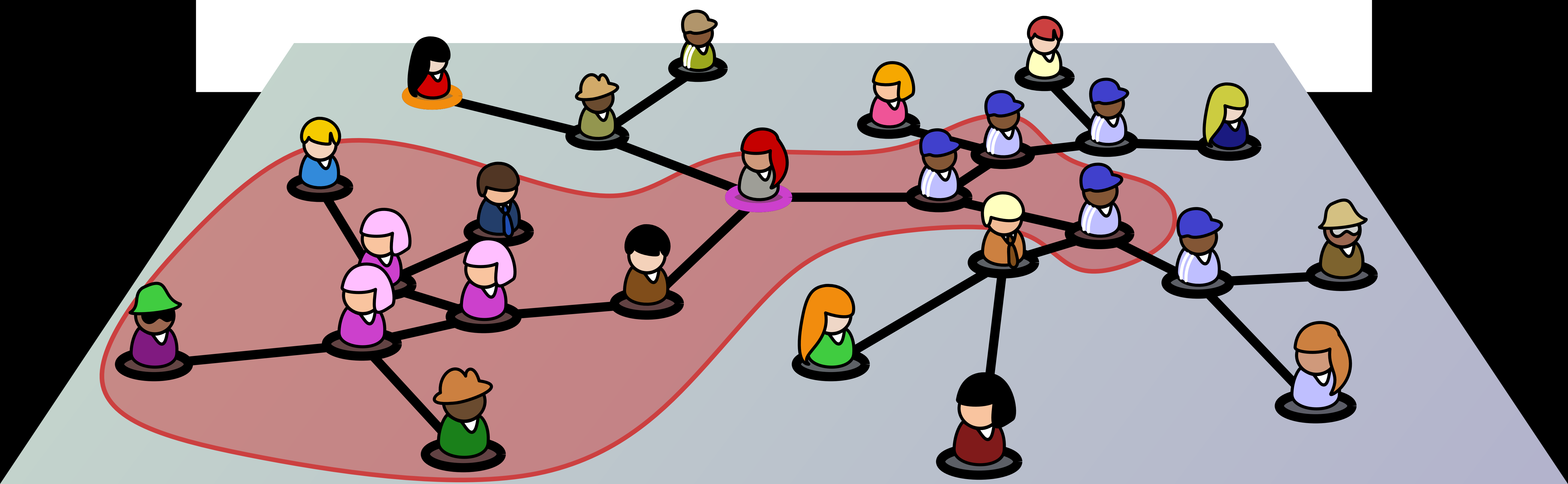
CATEGORY CONSTRUCTION

- Choose an arbitrary root for B
- For every node v in B and integer $d \leq \operatorname{diam}(B)$ create two categories
  - One contains all nodes in the left subtree of B at v and the first d levels of the right subtree of B at v



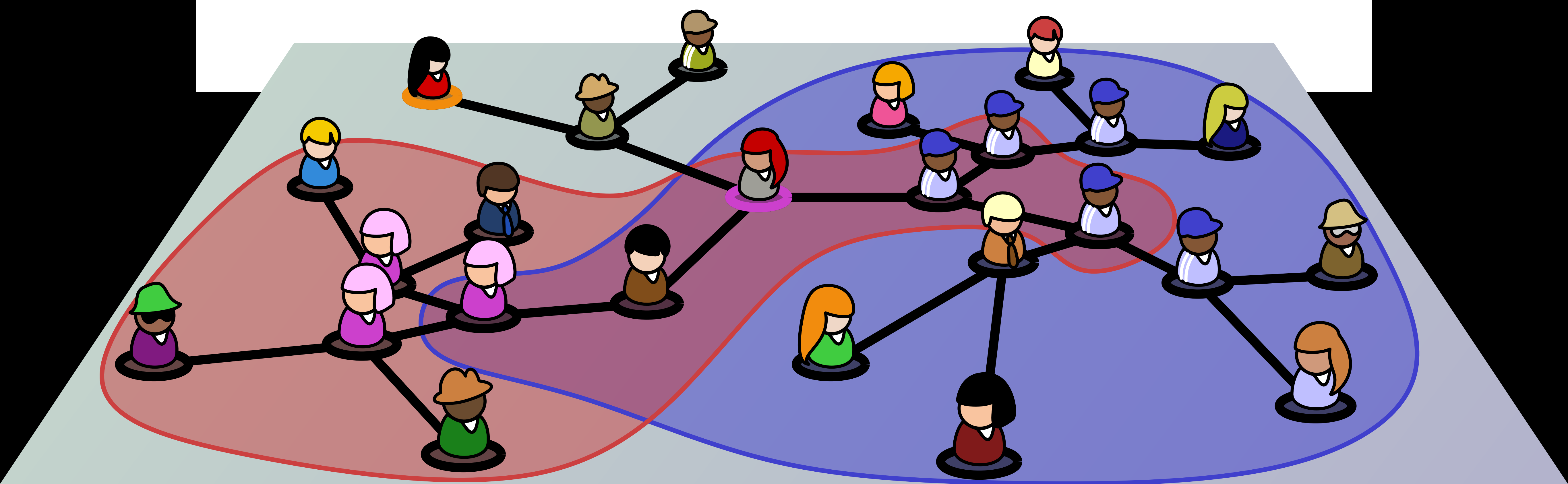
CATEGORY CONSTRUCTION

- Choose an arbitrary root for B
- For every node v in B and integer $d \leq \mathrm{diam}(B)$ create two categories
  - One contains all nodes in the left subtree of B at v and the first d levels of the right subtree of B at v
  - The other one is symmetric



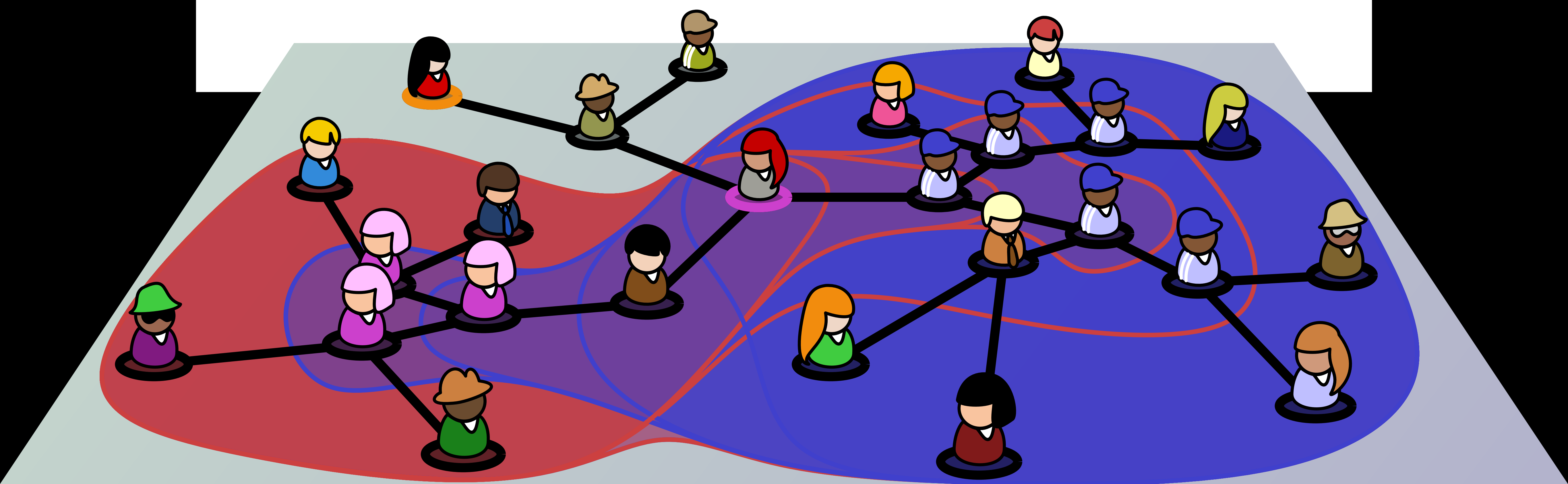
CATEGORY CONSTRUCTION

- Choose an arbitrary root for B
- For every node v in B and integer $d \leq \mathrm{diam}(B)$ create two categories
  - One contains all nodes in the left subtree of B at v and the first d levels of the right subtree of B at v
  - The other one is symmetric



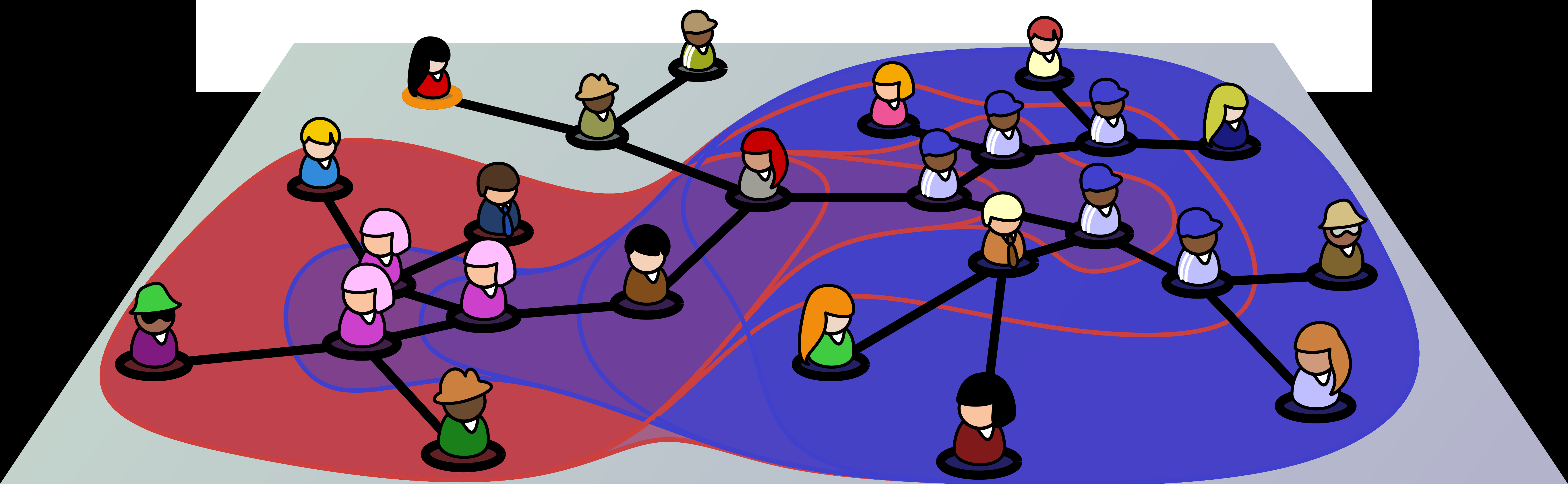
CATEGORY CONSTRUCTION

- Choose an arbitrary root for B
- For every node v in B and integer $d \leq \mathrm{diam}(B)$ create two categories
  - One contains all nodes in the left subtree of B at v and the first d levels of the right subtree of B at v
  - The other one is symmetric



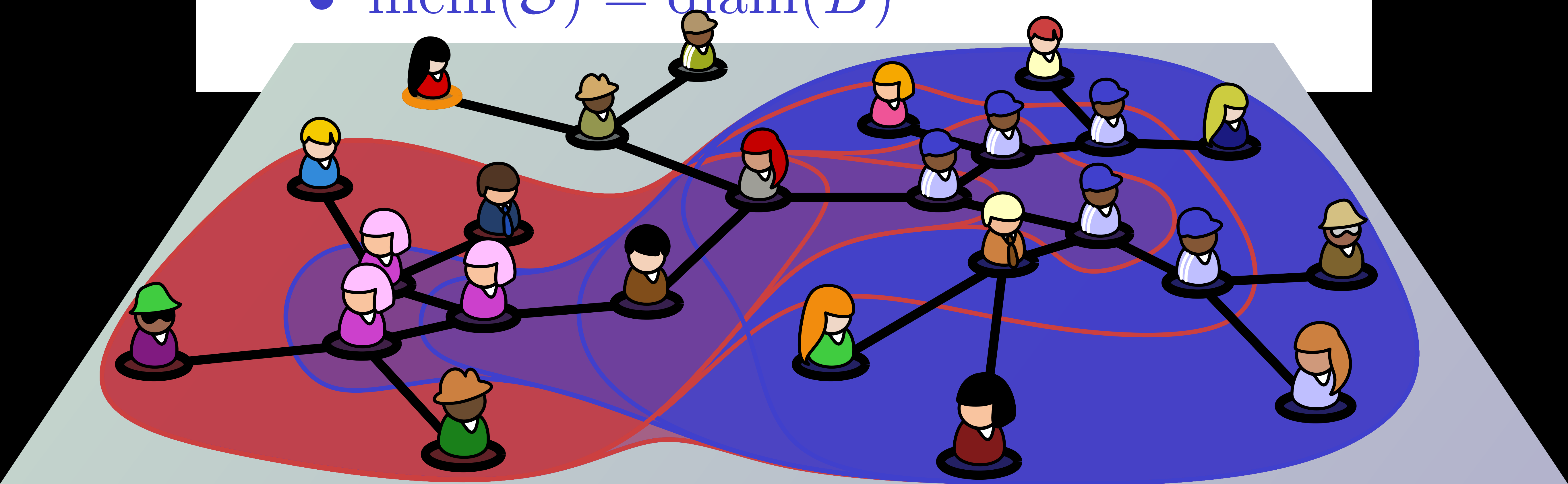
CATEGORY CONSTRUCTION

- Choose an arbitrary root for B
- For every node v in B and integer $d \leq \mathrm{diam}(B)$ create two categories
  - One contains all nodes in the left subtree of B at v and the first d levels of the right subtree of B at v
  - The other one is symmetric
- Routing works!



CATEGORY CONSTRUCTION

- Choose an arbitrary root for B
- For every node v in B and integer $d \leq \mathrm{diam}(B)$ create two categories
  - One contains all nodes in the left subtree of B at v and the first d levels of the right subtree of B at v
  - The other one is symmetric
- Routing works!
- $\mathrm{mem}(\mathcal{S}) = \mathrm{diam}(B)^2$



PART IV
DISCUSSION

CONCLUSION

CONCLUSION

- Main result

CONCLUSION

- Main result
  - For any given graph, there exists a set of categories of low membership dimension that makes simple routing work

CONCLUSION

- Main result
  - For any given graph, there exists a set of categories of low membership dimension that makes simple routing work
  - Theoretical evidence that category-based routing is a feasible explanation of Milgram's experiment

OPEN QUESTIONS

OPEN QUESTIONS

- Close the gap

OPEN QUESTIONS

- Close the gap
  - Membership dimension is between $\mathrm{diam}(G)$ and $(\mathrm{diam}(G) + \log n)^2$

OPEN QUESTIONS

- Close the gap
  - Membership dimension is between $\mathrm{diam}(G)$ and $(\mathrm{diam}(G) + \log n)^2$
- Real world data

OPEN QUESTIONS

- Close the gap
  - Membership dimension is between $\operatorname{diam}(G)$ and $(\operatorname{diam}(G) + \log n)^2$
- Real world data
  - To what extent are real data sets shattered and internally connected?

OPEN QUESTIONS

- Close the gap
  - Membership dimension is between $\operatorname{diam}(G)$ and $(\operatorname{diam}(G) + \log n)^2$
- Real world data
  - To what extent are real data sets shattered and internally connected?
- Slightly less simple routing

OPEN QUESTIONS

- Close the gap
  - Membership dimension is between $\mathrm{diam}(G)$ and $(\mathrm{diam}(G) + \log n)^2$
- Real world data
  - To what extent are real data sets shattered and internally connected?
- Slightly less simple routing
  - Can the routing strategy be made stronger in a fair way?

THANK YOU!