```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
import networkx as nx

# Load your data
file_path = 'Data Transfer requests permitted or Denied.csv'
data = pd.read_csv(file_path)

# Create a pivot table for the matrix visualization
pivot_table = data.pivot_table(index='From Person', columns='To Person', values='Permitted or not?', aggfunc=lambda x:
pivot_table = pivot_table.fillna('No Data')

# Replace string with numbers for heatmap
heatmap_data = pivot_table.replace({'Yes': 1, 'No': 0, 'No Data': None})

# Create a heatmap using Plotly
fig_heatmap = px.imshow(heatmap_data, text_auto=True, aspect='auto', color_continuous_scale='RdYlGn')
fig_heatmap.update_layout(title='Data Transfer Permission Matrix', xaxis_nticks=36)
fig_heatmap.show()

# Initialize the graph
G = nx.Graph()

# Add nodes and edges based on heatmap_data
for from_person in heatmap_data.index:
    for to_person in heatmap_data.columns:
        permission = heatmap_data.loc[from_person, to_person]
        if pd.notna(permission):
            if permission == 1:
                G.add_edge(from_person, to_person, color='green', style='solid', width=2)
            elif permission == 0:
                G.add_edge(from_person, to_person, color='red', style='solid', width=2)
            else:
                G.add_edge(from_person, to_person, color='blue', style='dot', width=2)  # For 'No Data'

# Use the same layout as before
pos = nx.spring_layout(G, seed=42)

# Create edge traces
edge_traces = []
for edge in G.edges(data=True):
    x0, y0 = pos[edge[0]]
    x1, y1 = pos[edge[1]]
    color = edge[2]['color']
    style = edge[2]['style']
    width = edge[2]['width']
    edge_trace = go.Scatter(
        x=[x0, x1, None], y=[y0, y1, None],
        line=dict(width=width, color=color, dash=style),
        hoverinfo='none',
        mode='lines')

    edge_traces.append(edge_trace)

# Create node trace
node_trace = go.Scatter(
    x=[pos[node][0] for node in G.nodes()],
    y=[pos[node][1] for node in G.nodes()],
    text=list(G.nodes()),
    mode='markers+text',
    hoverinfo='text',
    marker=dict(
        showscale=False,
        color='blue',
        size=20,  # Increased node size
        line_width=2),
    textposition="top center")

# Create the figure
fig_network = go.Figure(data=edge_traces + [node_trace],
                        layout=go.Layout(
                            title='<br>Organized Network Graph of Data Transfer',
                            titlefont=dict(size=16),
                            showlegend=False,
                            hovermode='closest',
                            margin=dict(b=20, l=5, r=5, t=40),
```

```
                  xaxis=dict(showgrid=False, zeroline=False, showticklabels=False),
                  yaxis=dict(showgrid=False, zeroline=False, showticklabels=False),
                  height=600))
```

```
fig_network.show()
```