In [1]:

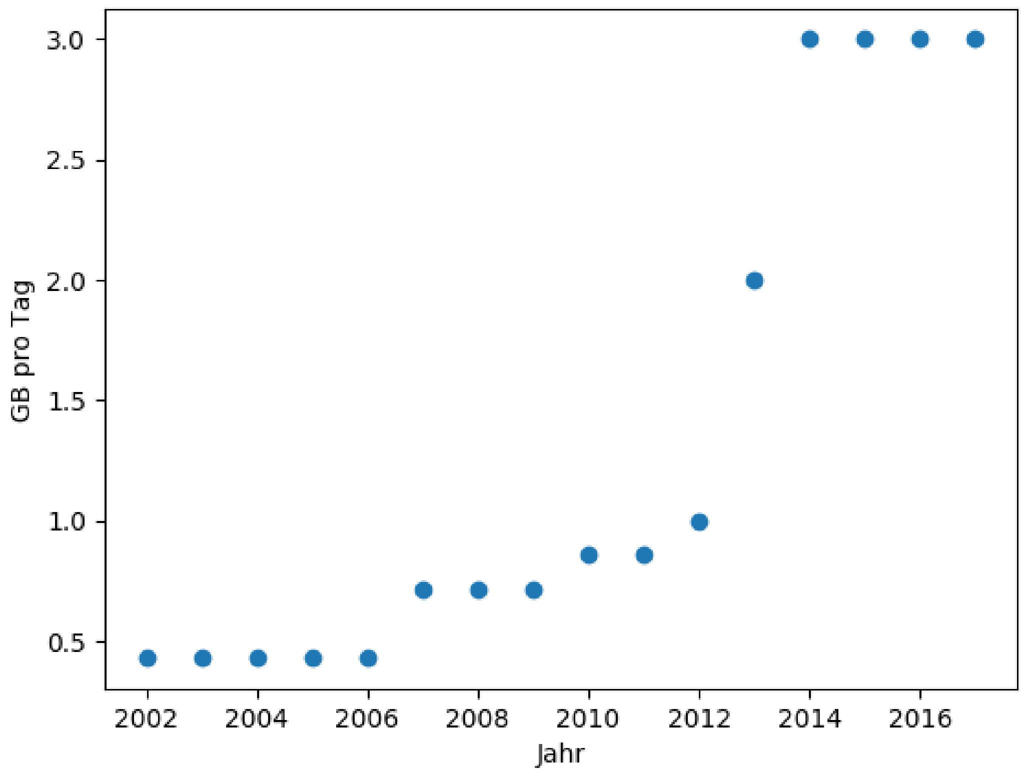
```
import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import curve_fit
%matplotlib notebook
```

In [2]:

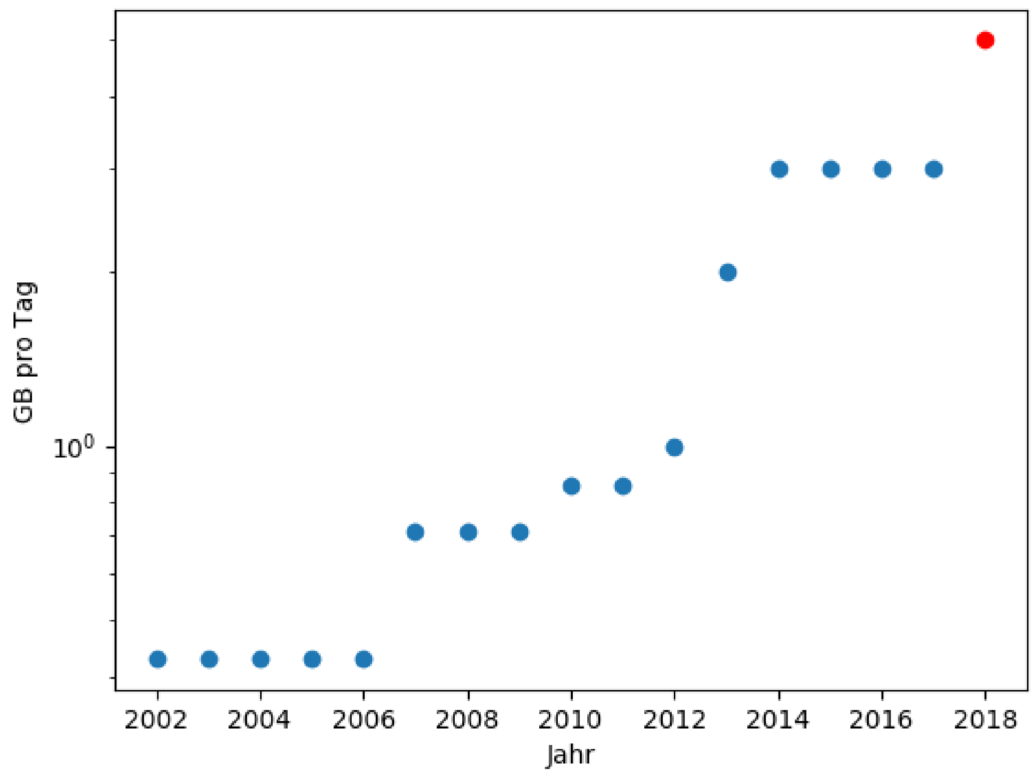
```
history = [
    [2002, 3/7],
    [2003, 3/7],
    [2004, 3/7],
    [2005, 3/7],
    [2006, 3/7],
    [2007, 5/7],
    [2008, 5/7],
    [2009, 5/7],
    [2010, 6/7],
    [2011, 6/7],
    [2012, 1],
    [2013, 2],
    [2014, 3],
    [2015, 3],
    [2016, 3],
    [2017, 3],
]
history = np.transpose(history)
years = history[0]
limits = history[1]
```

In [3]:

```
fig, ax = plt.subplots()
ax.plot(years, limits, marker='o', ls='None')
ax.set_xlabel("Jahr")
ax.set_ylabel("GB pro Tag")
plt.show()
```



In [4]:
```
fig, ax = plt.subplots()
ax.plot(years, limits, marker='o', ls='None')
ax.set_xlabel("Jahr")
ax.set_ylabel("GB pro Tag")
ax.plot(2018, 5, 'ro')
ax.set_yscale("log")
plt.show()
```



In [5]:
```
def exp_growth(t, y0, A, τ):
    return y0 + A * 2**((t-2002)/τ)
```

In [6]:
```
years = np.append(years, [2018])
limits = np.append(limits, [5])
[[y0, A, τ], pcov] = curve_fit(exp_growth, years, limits)
perr = np.sqrt(np.diag(pcov))
print("y0 = {:.3f}, A = {:.3f}, τ = {:.3f}".format(y0, A, τ))
```

```
y0 = 0.110, A = 0.183, τ = 3.455
```

In [7]:
```
# Daten der Bundesnetzagentur, Datenvolumen Breitband in Festnetzen
# https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Allgemeine
s/Presse/Pressemitteilungen/JB2017ZDF.pdf
# Seite 6, Datenvolumen pro Nutzer und Monat in GB
data_BNetzA = [
    [2011, 17],
    [2012, 21],
    [2013, 29],
    [2014, 33],
    [2015, 47],
    [2016, 60],  # Prognose
]
data_BNetzA = np.transpose(data_BNetzA)
[[y0_BNetzA, A_BNetzA, τ_BNetzA], _] = curve_fit(exp_growth,
                                                  data_BNetzA[0],
                                                  data_BNetzA[1])
print("y0 = {:.3f}, A = {:.3f}, τ = {:.3f}".format(y0_BNetzA, A_BNetz
A, τ_BNetzA))
```

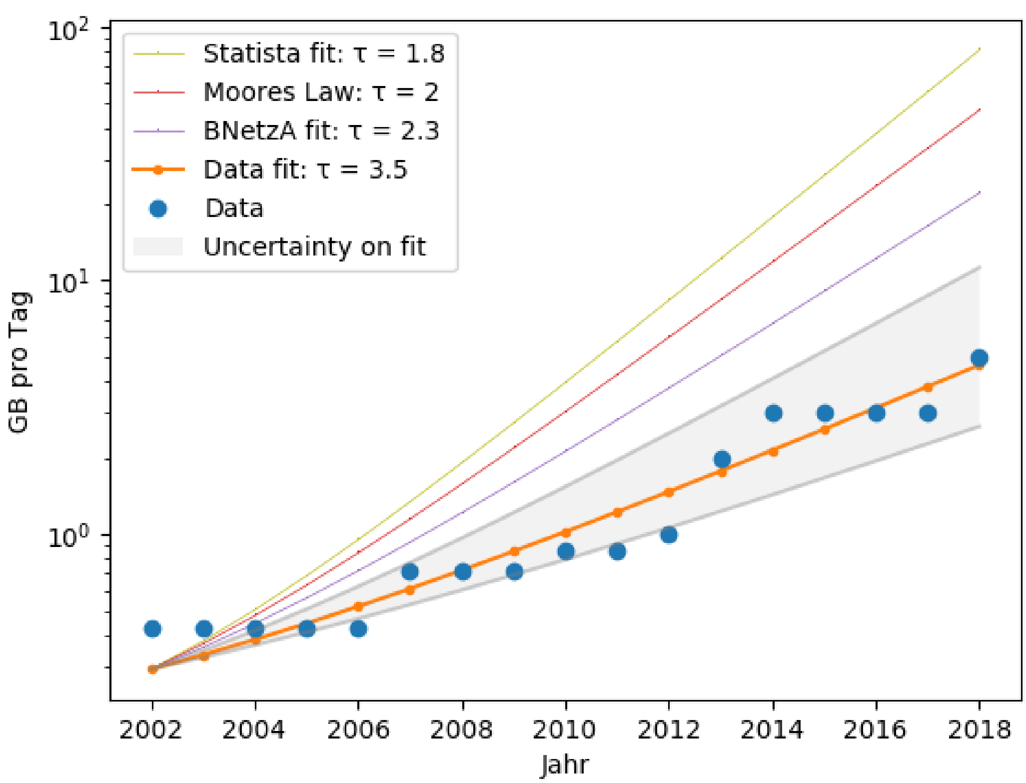
```
y0 = 4.627, A = 0.841, τ = 2.315
```

In [8]:
```
# Daten von Statista
# https://de.statista.com/statistik/daten/studie/3564/umfrage/durchsc
hnittliches-datenvolumen-pro-anschluss-seit-2001/
# Durchschnittliches Datenvolumen pro stationärem Breitbandanschluss
 und Monat in Deutschland von 2001 bis 2017 (in Gigabyte)
data_stat = [
    [2000, 1.8],
    [2002, 6.6],
    [2003, 8.8],
    [2004, 7.8],
    [2005, 6.5],
    [2006, 5.8],
    [2007, 6.9],
    [2008, 9.4],
    [2009, 10.1],
    [2010, 11],
    [2011, 12],
    [2012, 15],
    [2013, 20.9],
    [2014, 26.6],
    [2015, 44.2],
    [2016, 59.3],
    [2017, 79],   # Prognose
]
data_stat = np.transpose(data_stat)
[[y0_stat, A_stat, τ_stat], _] = curve_fit(exp_growth,
                                            data_stat[0],
                                            data_stat[1])
print("y0 = {:.3f}, A = {:.3f}, τ = {:.3f}".format(y0_stat, A_stat, τ
_stat))
```

```
y0 = 5.564, A = 0.249, τ = 1.820
```

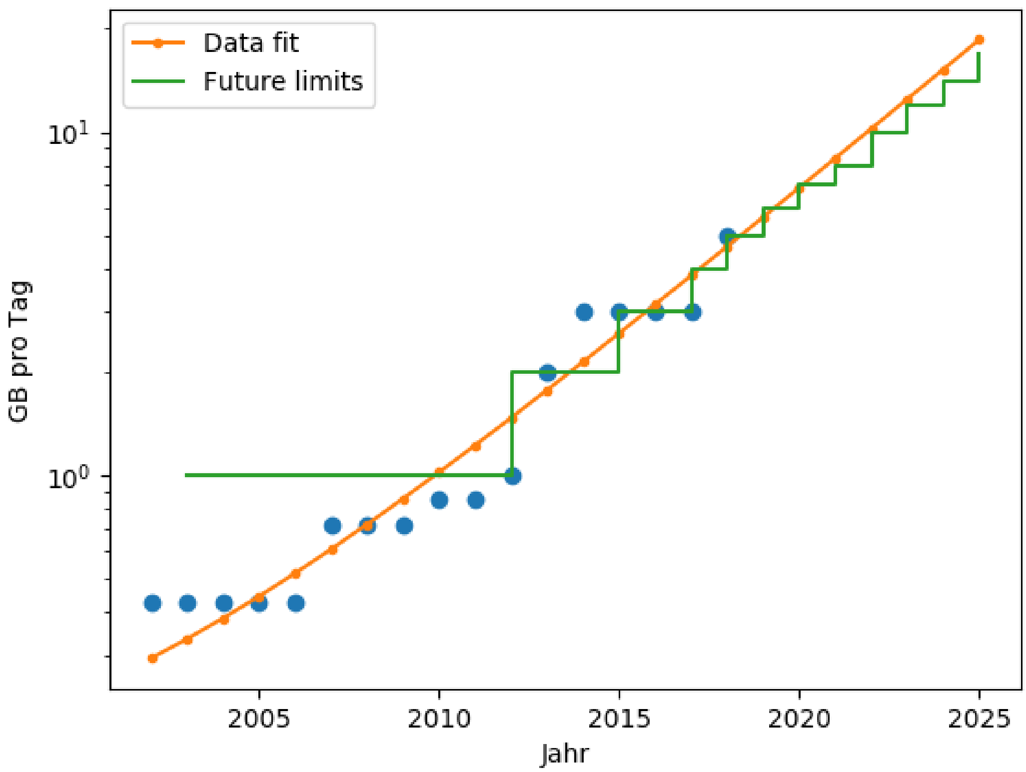
In [9]:
```
# Moore's Law
# N_Transistors ~ 2**(year/2) = e**(year/(2/ln(2))) = e**(year/2.9)
```

In [10]:
```
fig, ax = plt.subplots()

ax.plot(years, exp_growth(years, y0, A, τ_stat) , marker=',', c='C8',
        label='Statista fit: τ = {:.1f}'.format(np.round(τ_stat, deci
mals=1)), lw=0.5)
ax.plot(years, exp_growth(years, y0, A, 2) , marker=',', c='C3',
        label='Moores Law: τ = 2', lw=0.5)
ax.plot(years, exp_growth(years, y0, A, τ_BNetzA) , marker=',', c='C
4',
        label='BNetzA fit: τ = {:.1f}'.format(np.round(τ_BNetzA, deci
mals=1)), lw=0.5)
ax.plot(years, exp_growth(years, y0, A, τ) , marker='.', c='C1',
        label='Data fit: τ = {:.1f}'.format(np.round(τ, decimals=1)))

ax.plot(years, limits, marker='o', c='C0', ls='None', label='Data')
ax.plot(years, exp_growth(years, y0, A, τ+perr[2]) , marker='None', c
='grey', alpha=0.4)
ax.plot(years, exp_growth(years, y0, A, τ-perr[2]) , marker='None', c
='grey', alpha=0.4)
ax.fill_between(years, exp_growth(years, y0, A, τ+perr[2]), exp_growt
h(years, y0, A, τ-perr[2]),
                facecolor='grey', alpha=0.1, label='Uncertainty on fi
t')

ax.set_xlabel("Jahr")
ax.set_ylabel("GB pro Tag")
ax.set_yscale("log")
ax.legend()
plt.show()
```



In [11]:
```python
def agreement(t):
    return np.floor(0.8 + 0.18*2**((t-2002)/3.5))
```

In [12]:
```python
years_2025 = np.append(years, range(2019, 2026))
fig, ax = plt.subplots()
ax.plot(years, limits, marker='o', ls='None')
ax.plot(years_2025, exp_growth(years_2025, y0, A, τ) , marker='.', la
bel='Data fit')
ax.step(years_2025, agreement(years_2025), marker=',', where='post',
label='Future limits')
ax.set_xlabel("Jahr")
ax.set_ylabel("GB pro Tag")
ax.set_yscale("log")
ax.legend()
plt.show()
```



In [13]:
```python
for [y, d] in np.transpose([years_2025, agreement(years_2025)])[16:]:
    print("{:4.0f} beträgt das Limit {:2.0f} GB/Tag".format(y, d))
```

```
2018 beträgt das Limit  5 GB/Tag
2019 beträgt das Limit  6 GB/Tag
2020 beträgt das Limit  7 GB/Tag
2021 beträgt das Limit  8 GB/Tag
2022 beträgt das Limit 10 GB/Tag
2023 beträgt das Limit 12 GB/Tag
2024 beträgt das Limit 14 GB/Tag
2025 beträgt das Limit 17 GB/Tag
```